CONTRACTOR: Marine Systems Engineering Laboratory
University of New Hampshire, Durham, NH

PRINCIPAL INVESTIGATOR: Robert W. Corell

PROJECT TITLE: Research and Engineering Design of a Microcomputer Controlled Unmanned Free Swimming Vehicle

CONTRACT NO.: Research supported by the U.S. Geological Survey, Department of the Interior, under USGS Contract No. 14-08-0001-18636

PERIOD COVERED: November 1, 1979 - October 31, 1980

REPORT NUMBER: Final Report

DATE OF SUBMISSION: November 1980

FIGURES

1.0 INTRODUCTION

Under Contract No. 14-08-0001-18636 funded by the U.S. Geological Survey, the Marine Systems Engineering Laboratory (MSEL) of the University of New Hampshire has been developing technology related to unmanned, untethered and autonomous underwater vehicles designed for inspection purposes.

Three years ago the U.S. Geological Survey, through the intermediary of NOSC, initiated studies for the development of technology related to unmanned, untethered vehicles for offshore inspection purposes. The Geological Survey is charged with responsibility relating to the safety, as well as the environmental consequences, of pipelines and structures in the Coastal Zone. Since many of the locations requiring inspection imply danger for human divers, and pose a threat of entanglement for tethered vehicles, studies of technologies related to the ultimate development of autonomous vehicles to replace tethered vehicles in some work environments were initiated at both NOSC and at MSEL-UNH.

The use of intelligent vehicles for pipeline inspection implies an ability to locate the pipe, and the intelligence to follow it on a track that is essentially single dimensioned. Systems of sensing, required for the inspection process, methods of data recording or transmission are needed, and long range navigation must be provided.

When the inspection of offshore structures is contemplated, the technical tasks become much more sophisticated. The vehicle must transit to the structure, be able to move within its confines in three dimensions, and to locate, with precision, work stations of interest. Limited tasks, such as photography, sample collection, electric potential measurement are contemplated. More active tasks such as cleaning a weld have been discussed, although it is recognized that in a self-contained vehicle power is very limited. In either case difficult problems of technology are apparent. Their solution should have impact on developments throughout the inspection field.

Implicit in the studies was the need for an open-frame test vehicle to support the evaluation of developmental systems. An Experimental Autonomous Vehicle, called EAVE, carrying the suffix East to distinguish it from a related vehicle under study at NOSC, has been developed. The MSEL vehicle, as it appeared for the 1979 test series in Lake Winnipesaukee, is shown in Figure 1. It is seen to have thruster pairs in each axis, and buoyancy, electronics and power containers. The frame construction is designed for easy installation of experimental systems. In the referenced Figure the vehicle carries the sensor ring used for pipeline detection.

The Project Report for 1978 - 1979 describes the development of the system through the completion of the pipeline follower testing. This report continues with the modification of the vehicle system from a mission of pipeline following to structural inspection.

Figure 1

UNH VEHICLE

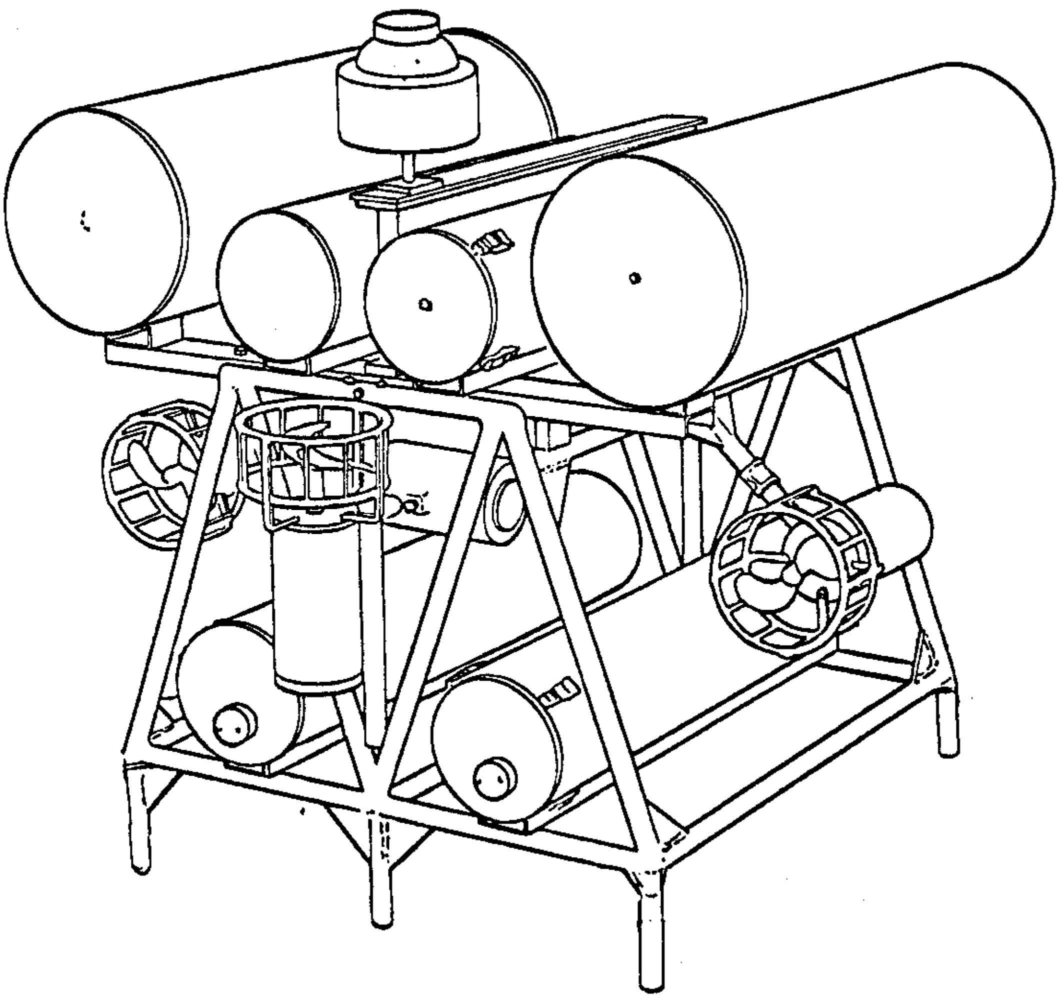
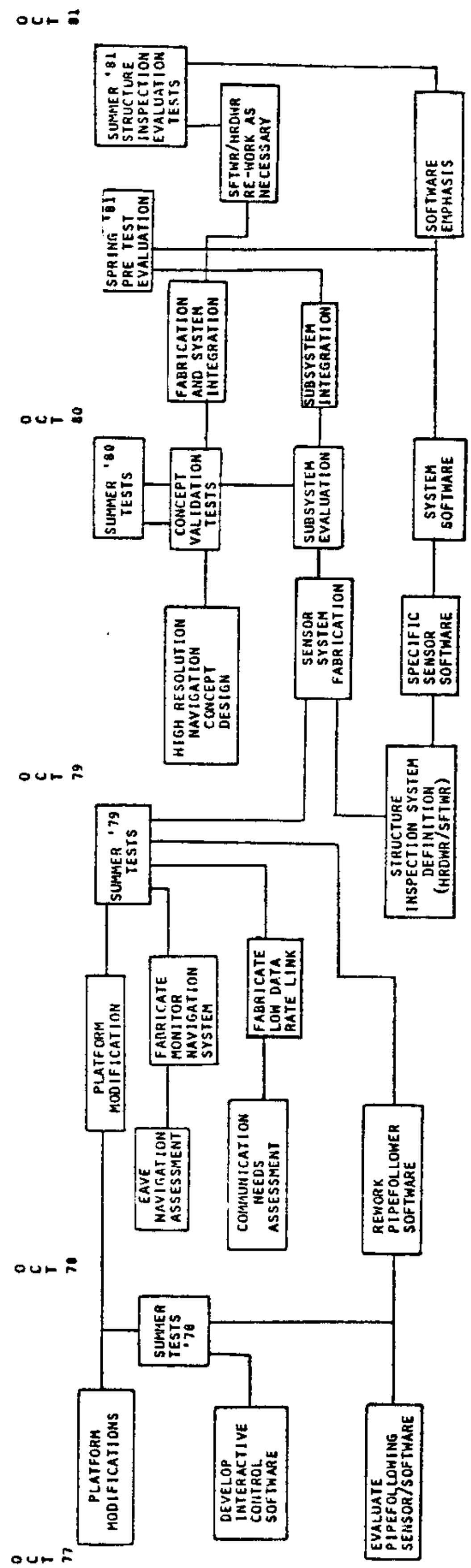


Figure 2



PROPOSED MILESTONES EAVE EAST

3.0 TECHNOLOGY SUMMARY

The task of designing an autonomous, untethered, unmanned vehicle, capable of successfully penetrating the confines of an offshore structure, maneuvering through a maze-like configuration, performing some useful tasks and returning safely, is difficult. The focus of the following material is on the technology of accomplishing that mission.

Five tasks were assigned to MSEL in the 1979-1980 work statement. They may be summed as:

- A conceptual study of the problems associated with structural inspection was to be made.

- A microprocessor-controlled navigation system would be designed and evaluated.

- The computer hardware and software that must be generated to conduct autonomous inspection of structures was to be identified.

- A software system would be devised that would accommodate a multi-processor computer, that would serve as an effective operating system, and that would address the problem of operator system communication.

4.0 STRUCTURE INSPECTION APPROACH USING AN UNMANNED, UNTETHERED MICROPROCESSOR CONTROLLED SUBMERSIBLE

Structure inspection in greater ocean depths presents many hazards undesirable for manned missions. The use of an unmanned vehicle is therefore highly desirable. Past usage of a tether has demonstrated the hazards of entanglement. The problems encountered in the navigation of an unmanned vehicle through a structure are many but not insurmountable. They are examined in their study, however.

The proposed solution to untethered vehicle navigation was formulated within the constraints of this particular vehicle's capabilities. The vehicle used in this case is a microprocessor based machine, with a navigation system which utilizes fixed bottom mounted acoustic transponders. The limitations of this system are its low computational power, and its inability to sample its environment for information. The system does however allow for navigation within a fixed coordinate system.

When first encountered, the problem of autonomous inspection seems to demand certain high level capabilities. It makes sense that the vehicle should be completely autonomous with the ability to make decisions based on its environment, and be able to navigate from place to place using this decision making power.

This approach is capable of extension as intelligence expands in the vehicle. If sensors are placed on the vehicle, either tactile or acoustic, the relative position of vehicle to a structure may be measured, instead of calculated. This step is beyond the scope of the current program, but is eminently practical. It is clear further that if a complete sensor system were installed, a rather complex step, then the vehicle could generate a map of unknown structure as it proceeds.

Given current limitations the problem of structure inspection becomes one of rigorously defined missions where the vehicle does a minimum of computation. The problem of structure inspection then becomes one of structure analysis. The process that would take place for a particular mission would be as follows:

1) "Analyze" the structure on a high powered computer.

2) Define the mission in terms of nodes and operations to be performed, referenced to the coordinates of the navigation system.

3) Generate the series of commands to perform the mission.

4) Load the vehicle with the command list.

5) Put it in the water and initiate the mission.

1) Structure Analysis

This phase of the mission definition requires that the structure to be inspected be defined completely. This definition consists of lists of vertices, members (beams), and planar polygons that make up the structure. The input of this information in the case of more complex structures would be graphically assisted. Once provided with a description, the analysis program produces a map or node network that defines all safe paths through the structure (Figure 3). The techniques used in this analysis are general enough that they may be applied to nearly any under water structure.

The structure analysis process goes through several levels of information generation in the creation of the safe path network. The first step in the analysis is the definition of what are referred to as passage frames. Passage frames are those polygons defined by traversing the perimeter of all structure polygons at a safe distance to their interior (Figure 4). The definition of these polygons is done in a series of steps which are performed for each polygon is the structure.

First the polygon is transformed into the X-Y plane (Appendix A). Then the program must determine whether the interior of the polygon is on the right or left as the polygon is traversed (Appendix A). For every side in the polygon a series of points forming a line parallel to that side are generated.

Figure 3

STRUCTURE NODE NETWORK

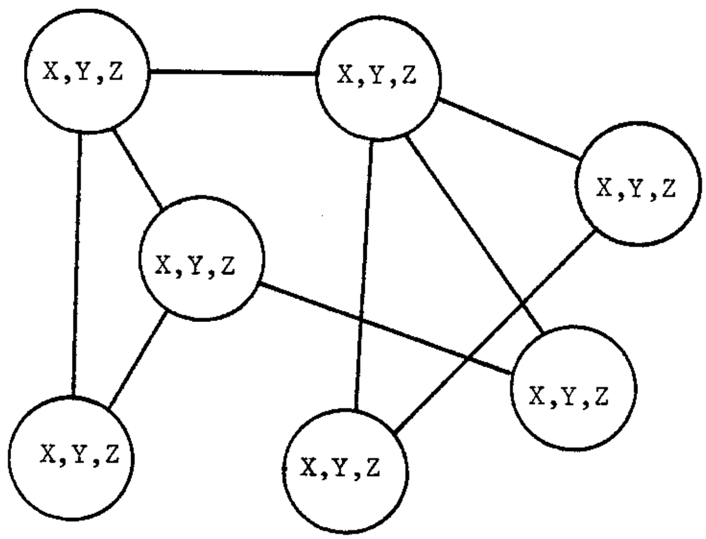
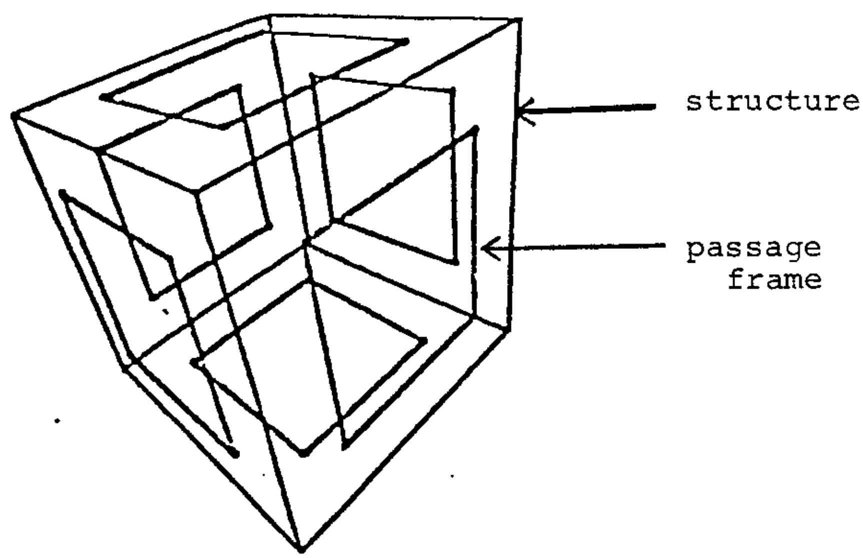


Figure 4

PASSAGE FRAMES FOR A CUBICAL STRUCTURE



These points are found by translating the side so that the first end point is at the origin and then rotating a point on the ray defined by the side 90 degrees to the interior of the polygon. In order that the point be a safe distance from the endpoint it is located a safe distance from the endpoint on the ray before it is rotated. Once rotated the side and rotated point are both translated back into position. The other endpoint of the line to be traversed is found in the same fashion, by translating the second endpoint to the origin and rotating a point or the ray and translating back. Once the end points of the line to the interior are found it may be traversed for safety. A finite evenly distributed number of points are checked along each side scanned. While scanning each point is checked against all the other members in the structure. (See Appendix A for distance calculations). Nodes and paths are defined based on the following rule:

> If a location is determined to be unsafe then a check is made to see if the previous location was safe. If it was then a node is generated at that location and a path is defined from that node to the previous node.
>
> If a location is determined to be safe then a check is made to see if the previous location was unsafe. If it was then a node is generated at the current location.

The first and last points on a line always generate nodes if they are safe locations. These special cases can be thought of in terms of the point previous to the first is unsafe and the point following the last is unsafe.

Whenever nodes are generated a check is made to insure that no two nodes have the same location.

Once each side of each polygon has been processed a series of passage frames are defined in terms of nodes and paths. The nodes generated thus far may be augmented by the user to ensure that the network include nodes at particular locations. The user may find this desirable for mission start and end locations and also for defining locations where some mission operation might occur.

At the next level of analysis all the nodes generated thus far are checked for possible inter connection. Each possible path is scanned in the same fashion as the interior lines were when defining passage frames. The safe and unsafe nature is handled in an identical fashion, generating more nodes and paths where appropiate.

As nodes are added to the network they are also checked for possible interconnection. In most structural cases some sort of artificial limit must be placed on this process. A convenient and easy to implement limit is to allow only a certain number of nodes to be generated. This number should be some function of the complexity of the structure.

2) <u>Mission High Level Definition</u>

This stage of the definition requires that the operator define the mission using high level commands such as MOVE NODE#.

3) <u>Command Generation</u>

Command generation uses the high level command list, the structure map, and an initial and final position to generate a list of low level commands. These commands will be generated such that the vehicle will stay on only safe paths as defined by step 1. The routine which would produce such a list would use techniques from A.I. Similar to the traveling salesman problem where the least costly path from one place to another is found.

4,5) <u>Load and Go</u>

The vehicle is loaded with the list of instructions and placed at the start location. Once in position the mission is initiated and the vehicle simply follows its preprogrammed list of instructions.

The Test Mission: An Example

For the first actual test of the vehicle's ability to perform these maneuvers it will navigate through a simple structure taking pictures of predefined points on the structure. The following is a description of the process that will take place in preparation for the mission.

As described on page 12 the process starts with the description of the structure. The test structure illustrated in Figure 5 will be described for input to the structure mapping program. The description consists of three lists, a list of vertices, a list of members, and a list of polygons. The first list defines all points needed to define the structure. The second list uses the first list's points to define all structural members. The third list uses the list of members to define all planar polygons in the structure. Following these lists is a start and finish location and the optional definition of operation locations, in the test case these will be the locations of the photographs. Figure 6 is the definition that will be used in the test.

Figure 5

TEST STRUCTURE

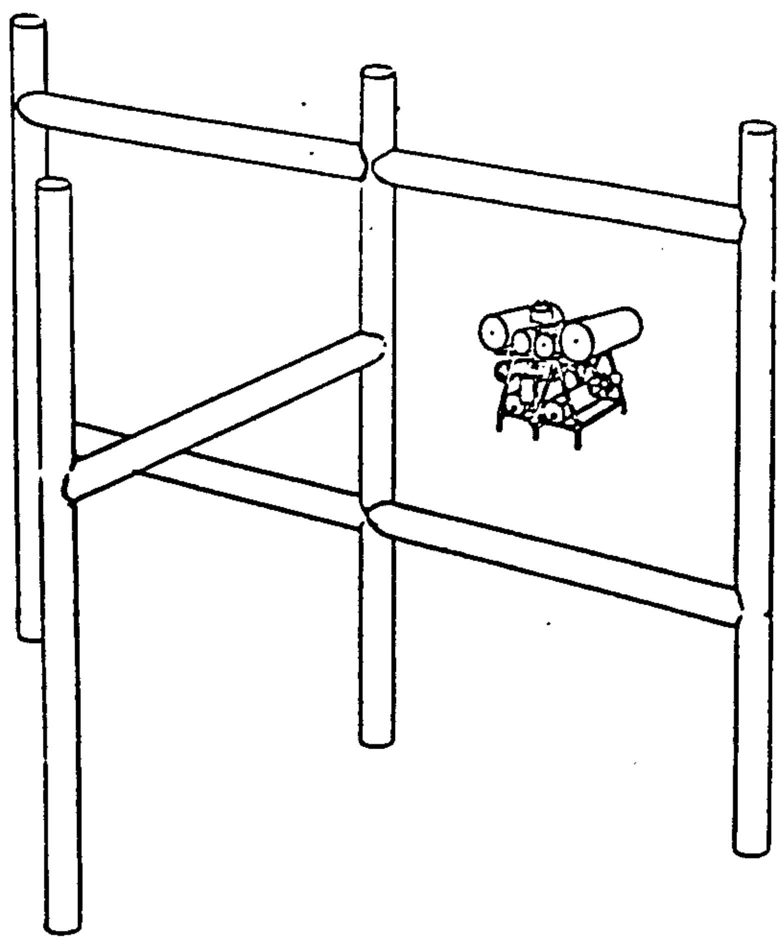


Figure 6

TEST STRUCTURE NODE NETWORK DATA

```
16                 The number of vertices that make up the structure
0.0 0.0 20.0
0.0 0.0 24.0
0.0 0.0 33.0
0.0 0.0 36.0
10.0 0.0 20.0
10.0 0.0 24.0
10.0 0.0 30.0
10.0 0.0 33.0      The list of vertices that make up the structure
10.0 0.0 36.0
20.0 0.0 33.0
20.0 0.0 24.0
20.0 0.0 20.0
20.0 0.0 36.0
10.0 10.0 30.0
10.0 10.0 20.0
10.0 10.0 36.0
15                 The number of structural members that make up
1 4                  the structure
2 3
2 6
3 8
5 9
6 8
6 11
8 10               The list of vertex pairs defining the structure
12 13
11 10
7 14
15 16
5 15
14 15
5 7
3                  The number of polygons that make up the structure
4                  The number of members
2 3 6 4            The list of members
4                                          The list of member lists
6 7 10 8                                   defining the structure's
4                                          polygons
13 14 11 15
10.0 -20.0 30.0    The start position
10.0 -20.0 30.0    The finish position
7.0 5.0 30.0       Operation position #1
13.0 5.0 30.0      Operation position #2
```

On having defined the structure the operator will "analyze" the structure using a structure mapping program on a Digital Equipment PDP-1Ø computer. The structure mapping program will take as input the structure description and produce as output a map of the structure. The structure maps produced by this program are node networks as illustrated in Figure 3. A portion of the actual data describing the node network for the test structure appears in Figure 6.

Once this network has been defined the bulk of the computation has been completed.

The second step in the process is to define the mission in terms of operations to be performed. The available operations are:

R# Rotate # degrees about z axis to x axis

M# Move to node #

P Take a picture at the current position and heading

The process of defining a mission through a simple structure is relatively simple. The mission description for our test appears in Figure 7.

Figure 7

MISSION DESCRIPTION

M 3	-	move to node 3
R Ø	-	rotate to Ø degrees from the x asix of the structure
P	-	take a picture at the current location and orientation
M 4	-	move to node 4
R 18Ø	-	rotate to 18Ø degrees from the x axis of the structure
P	-	take a picture at the current location and orientation

Note: The moves to the start and finish locations are generated automatically.

As the third step in preparation a command generation program reads the mission description and produces a complete list of commands to the vehicle such that it will carry out the mission in an efficient fashion.

The fourth step involves loading the command list into the vehicle's command computer at a predefined memory location. On completion of this step the mission preparation is complete. Once placed in the water near the structure the vehicle will carry out the mission and upon completion of the mission return to the surface for recovery.

It should be noted that when working with larger more complex structures such as the NDE structure all the same procedures apply. The added complexities of describing a larger structure and the complicated nature of the node network produced by such a structure make necessary the use of some operator aids. These aids consist of interactive graphics systems which are used in the definition of the structure and also in the definition of missions. With the use of operator aids, the problem of real life structure inspection is solvable.

Since the computational power required to deal with different structures increases nearly expodentially with structure complexity onboard computational power is a serious consideration. For a realistic application which requires navigation through a real life structure the vehicle requires high onboard computational power to do routine decision making and the vast quantity of calculation required to maneuver safely. For example in a comparison between analyzing the structure described in this paper and analyzing the NDE structure, the NDE structure requires between 30 and 40 times the calculations (processor time). Broken down, the NDE structure requires twelve times the passage frame calculations, which are costly in CPU time, and 3* 33!/12! times the number of path calculations, which are low cost in terms of CPU time.

(*The 3 refers to the number of structural sections.)

Possible Extensions:

Upon the establishment of a communication link with the vehicle, another degree of capability is created. With the use of a communication link missions can be controlled to a great extent by a ship or land based computer with operator interaction. In either the simple or extended vehicle the need for on board decision making concerning paths of travel is minimized.

Navigation in a Dynamic Environment:

After dealing with navigation in a known world one turns to the problems of navigation in unmapped areas. Although there is little similarity between static and dynamic environments some insights may be gleened from structure inspection work. There are also possible applications of structural navigation techniques in the unknown environment. For example, a mission which involved traveling through an unmapped area to a structure or form could rely completely on structure mapping techniques once the object is arrived at.

Static vs. Dynamic:

The major differences between the two problems is the need for continual sensory information about an unknown area, and the inability to do calculations prior to the mission. These differences make the two problems very distinct.

5.Ø EAVE's MULTIPROCESSOR SYSTEM

The vehicle's multiprocessor system is made up of four Intersil 61ØØ micros, (Figure 8), one primary mission controlling computer and three secondary computers that have more specialized functions. The mission controlling computer or command computer communicates with the three secondary computers receiving and distributing commands and data. The three secondary computers are the control computer, the navigation computer and the communication computer. The control computer receives both commands and position data from the command computer and uses them to control the vehicle's thrusters. The control computer also returns data and a command completed signal. The navigation computer receives data from the command computer and from an on-board acoustic transponder system and uses them to calculate position and heading which are returned to the command computer. The communication computer receives data from the command computer which is transmitted to the surface via an acoustic link and sends commands to the command computer which are received from the surface over the same link. Figures 8A an 8B illustrate physical details of the computer.

Figure 8

COMPUTER DIAGRAM

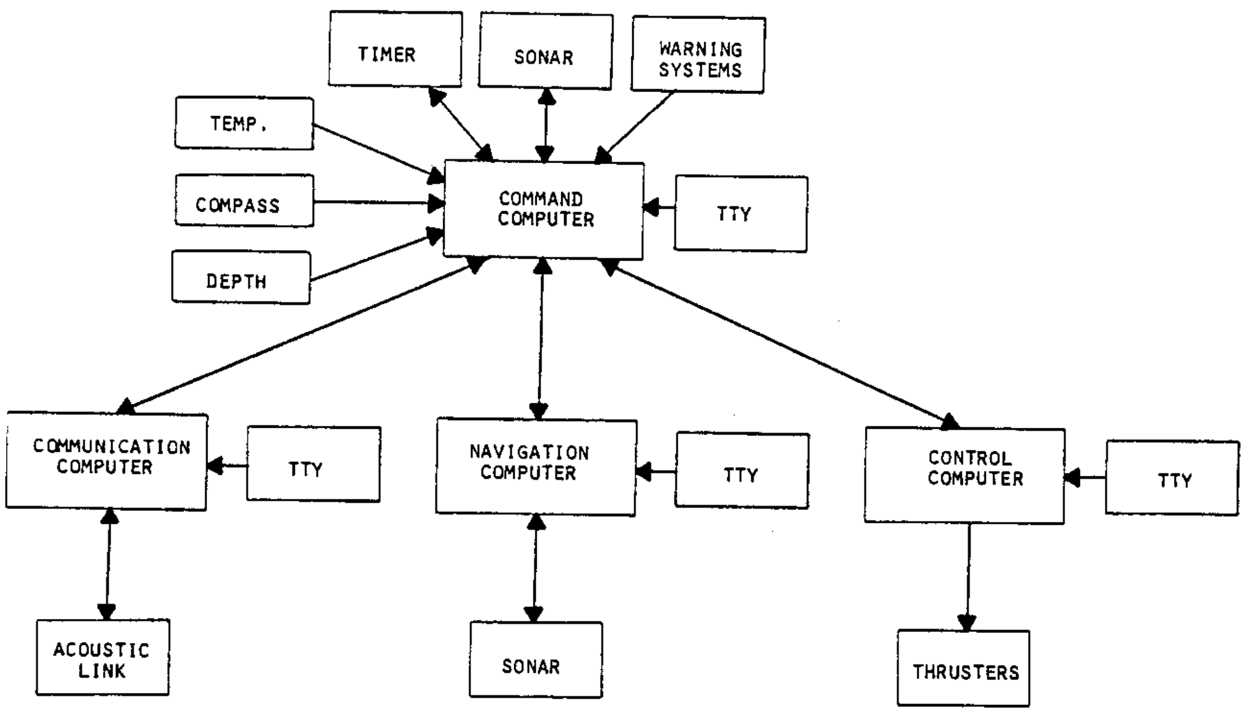
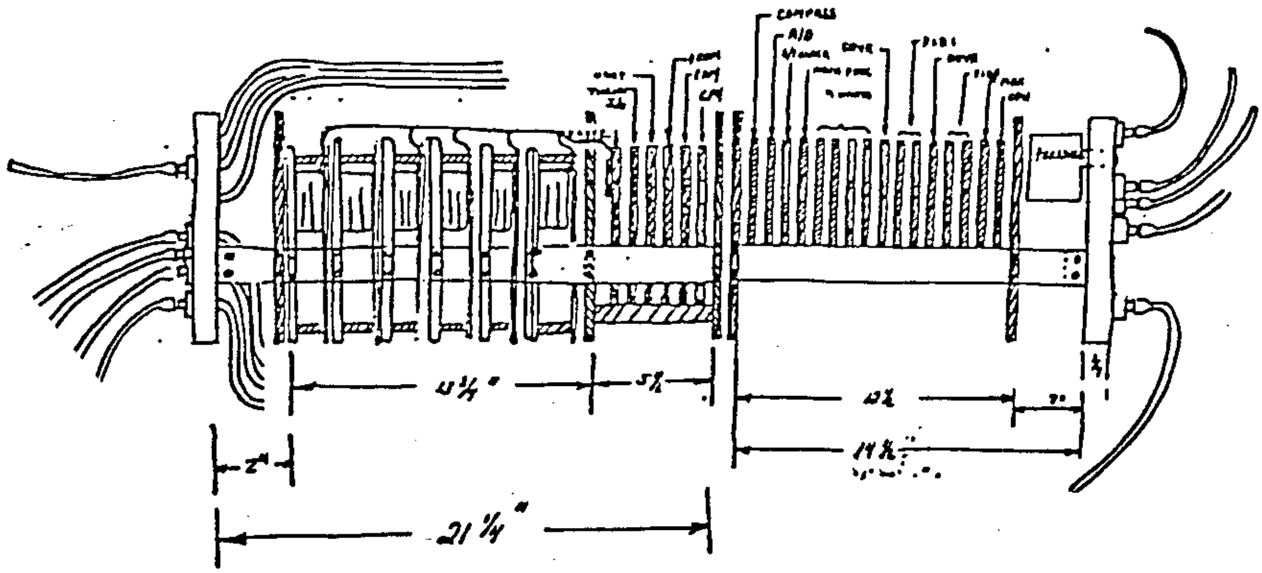


Figure 8A

DETAILS OF EAVE COMPUTER SYSTEM FABRICATION

(Preliminary)



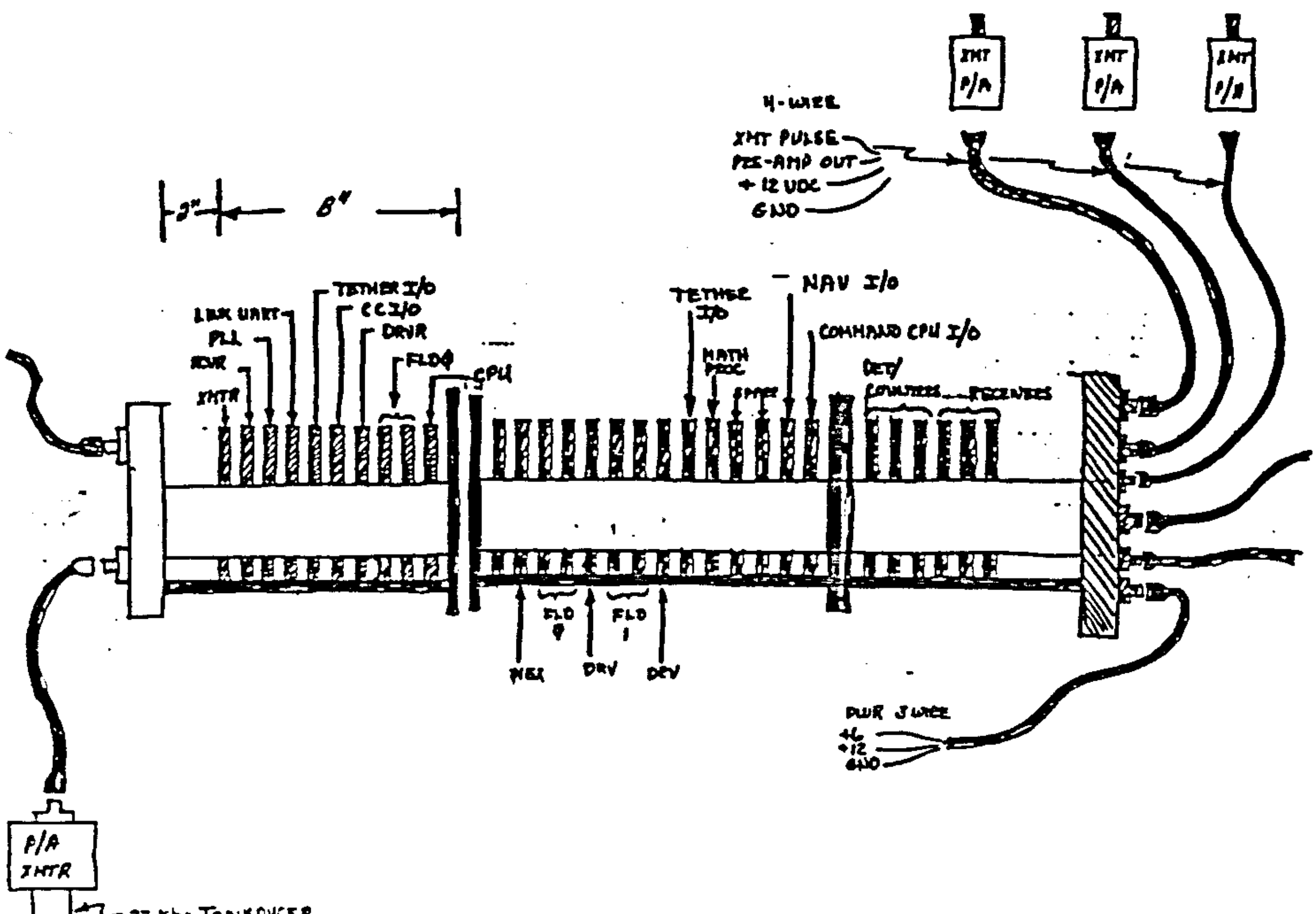
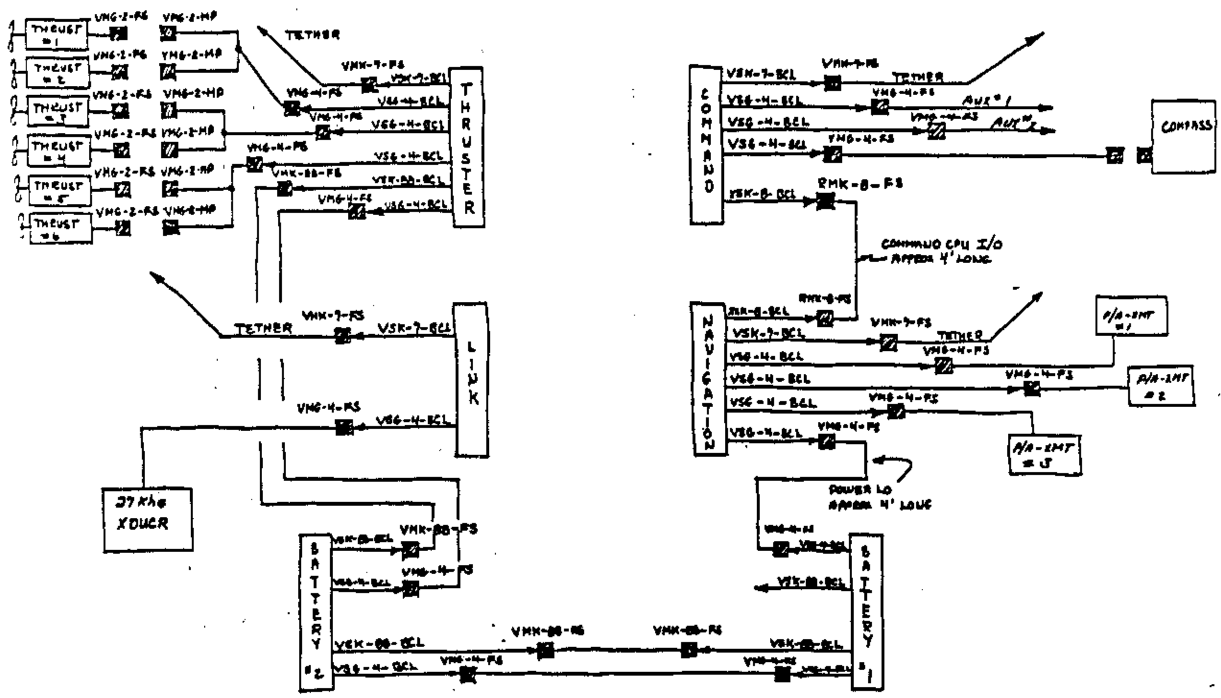


Figure 8B

DETAILS OF EAVE CONNECTOR DEFINITIONS,
ILLUSTRATING THE FUNCTIONS INCLUDED IN THE
STRUCTURAL INSPECTION SYSTEM

(Preliminary)



5.1 COMMAND COMPUTER

The command computer receives commands from the tether and the communication computer. Both of these are treated strictly as TTY's and may issue only standard commands listed in Appendix A. Once received, the command is then interpreted and action is taken. In addition to receiving and carrying out commands the command computer has a mission task which was set up prior to launch. The mission task simply interprets a list of commands stored in some list form. Lastly, the command computer is responsible for the gathering and distribution of data. Data is produced by the NAV and control computers and consumed by both of these and the communications computer. To maintain up-to-date data and allow ready access of it, a data base (common area) is created into which all new data is written and from which all data is read. Data is written into the data base by a update task and is read from it by defining channel output buffers in this common area. The State Diagram of the Command Computer is shown in Figure 9, while the data base is summarized in Figure 10.

Figure 9

STATE DIAGRAM OF COMMAND COMPUTER

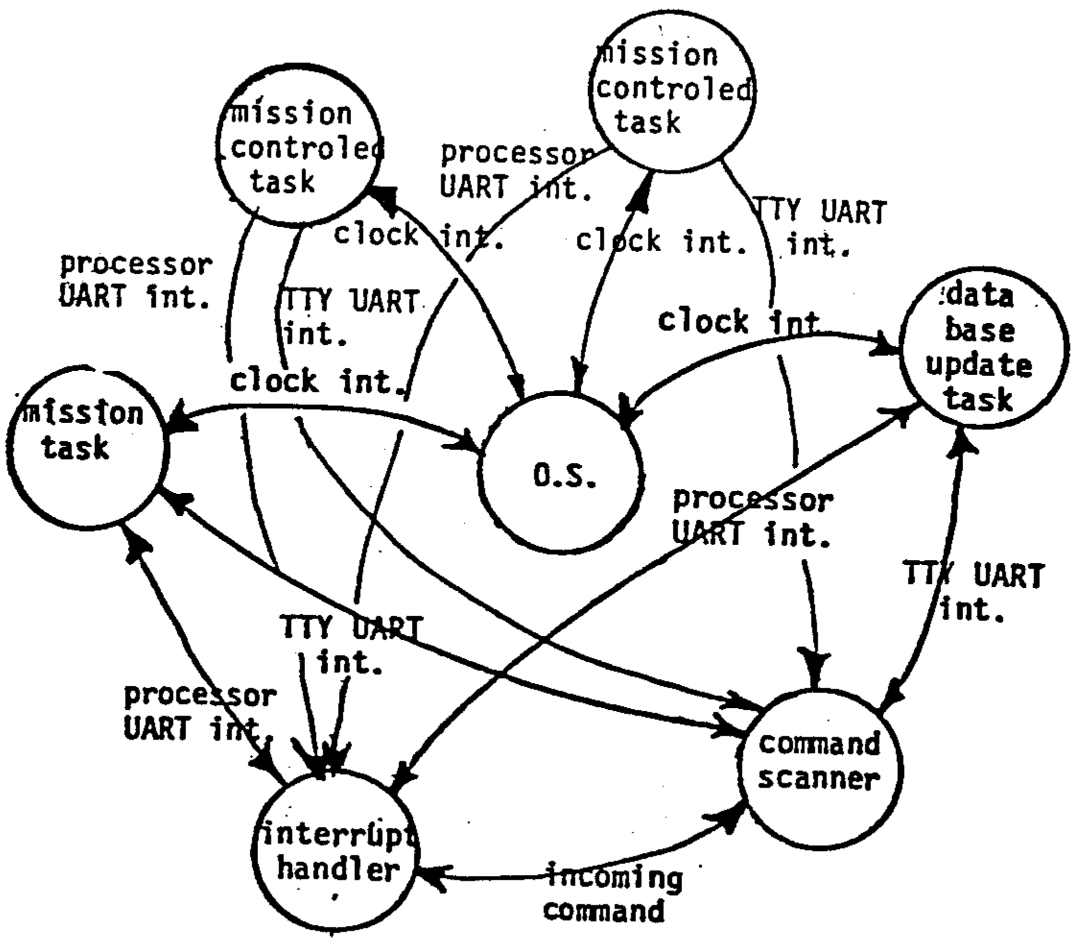


Figure 10

COMMAND COMPUTER DATA BASE

DB	1	reliability word	
	2	rel head	X
	3	range	Y
	4	rel head dot	abs head
	5	range dot	X dot
	6	---	Y dot
	7	---	head dot
	8	Z	
	9	Z dot	
	10	compass	
	11	TX	
	12	TY	
	13	TZ	
	14	TO	

5.1.1 COMMAND COMPUTER I/O CHANNEL DEFINITIONS

A. Command computer - NAV I/O channel 1

used by data base update task and NAV

in command computer the output buffer words are:

Buffstl	points to the eighth word in the command computer data base
Buffendl	points to the eleventh word in the data base (1Øth + 1)
Qstl	will almost always point to the eighth word in the buffer because reads will always be done in groups of three.
Qendl	will almost always point to the eighth word also because the buffer is almost always full.
#wordsl	will almost always be three because the buffer is almost always full.
#wantedl	will usually be zero because read requests should always be satisfied almost immediately.

UART addr holds the five bit address of the UART linking the command computer and the NAV computer.

in command computer the input buffer word are:

Read buffer pointer1	points to the first location in the command computer data base.
Buffer ready flag1	usually indicates not ready because the update task issues a new read request as soon as the last read is completed.
UART addr	holds the five bit address of the UART linking the command computer and the NAV computer.

B. Command computer - Control computer I/O channel 2

used by data base update task and Auto alt. and Travel tasks

in command computer the output buffer words are:

Buffst2	points to the first word in the command computer data base
Buffend2	points to the tenth word in the data base (9th + 1)
Qst2	will almost always point to the first word in the data base because reads will always be done in groups of seven.
Qend2	will always point to the first word because the buffer should almost always be full
#words2	will almost always be seven because the buffer is full and the control computer must be able to read at any time.
#wanted2	will usually be zero because all I/O requests should be handled immediately.
UART addr	holds the five bits address of the UART linking the command computer and the control computer.

in command computer the input buffer words are:

Read buffer pointer2	points to word eleven in the command computer data base.
Buffer ready flag2	usually indicates not ready because the update task issues a new read request as soon as the last read is completed.
UART addr	holds the five bit address of the UART linking the command computer and the control computer.

C. Command computer - Control computer I/O channel 3
used by mission task and Travel task

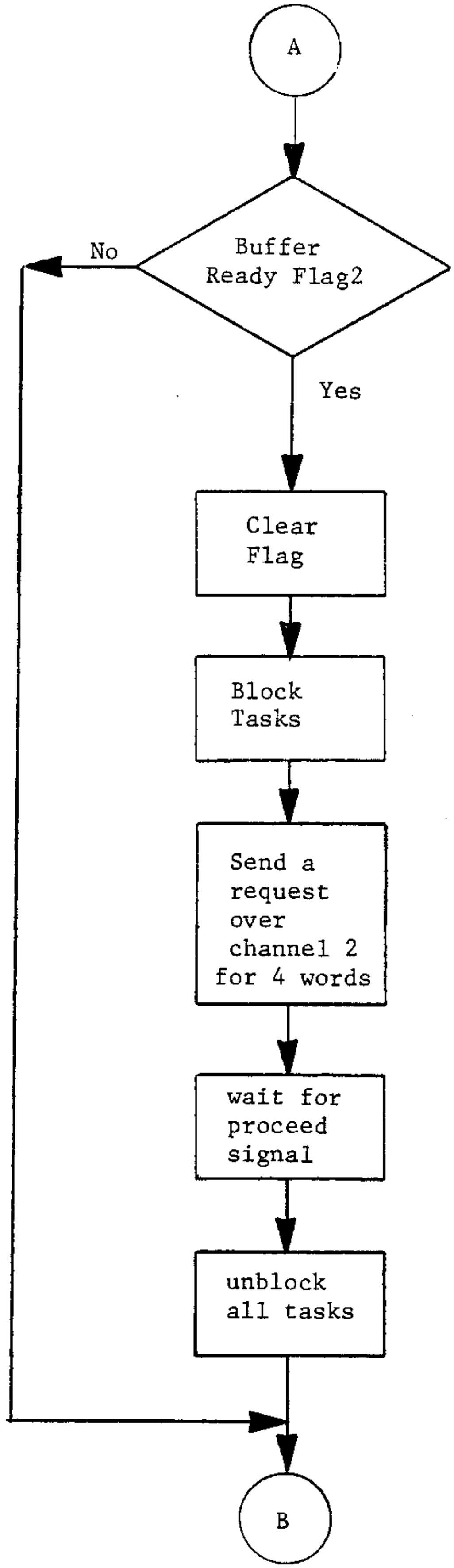
in command computer the input buffer area words are:

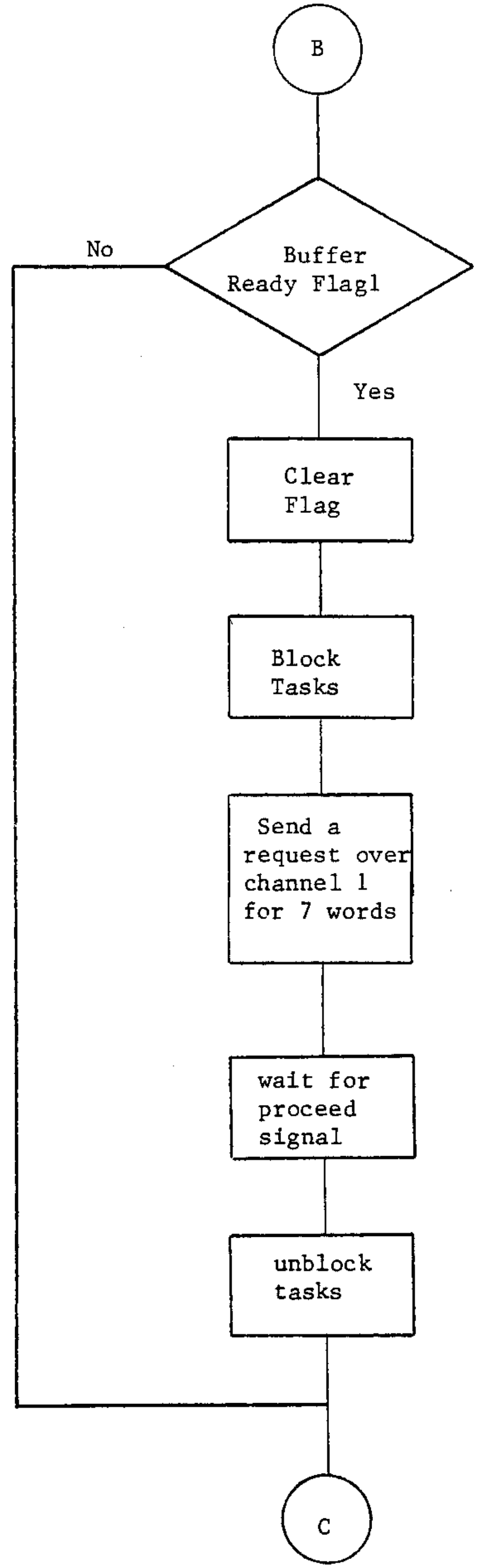
Read buffer pointer3	points to the done flag being checked by the mission task.
Buffer ready flag3	usually indicates not ready because more time is taken up by the control computer carrying out commands than is taken by the command computer to get a new command.
UART addr	holds the five bit address of the UART linking the command computer and the control computer.

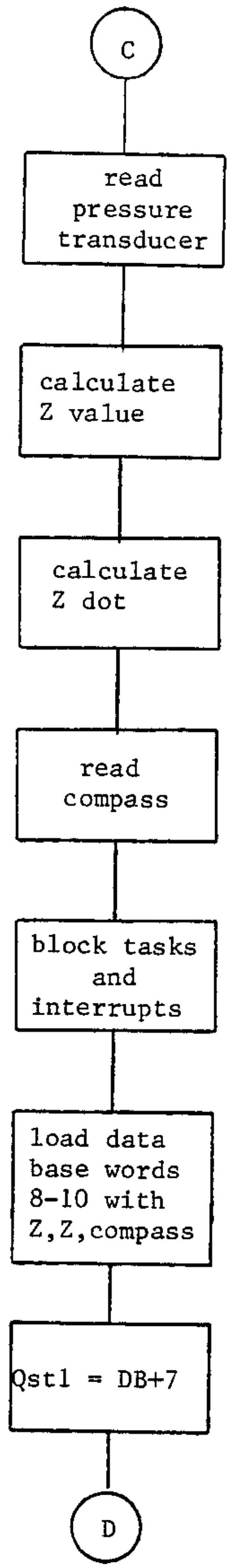
5.1.2 COMMAND COMPUTER DATA BASE UPDATE TASK

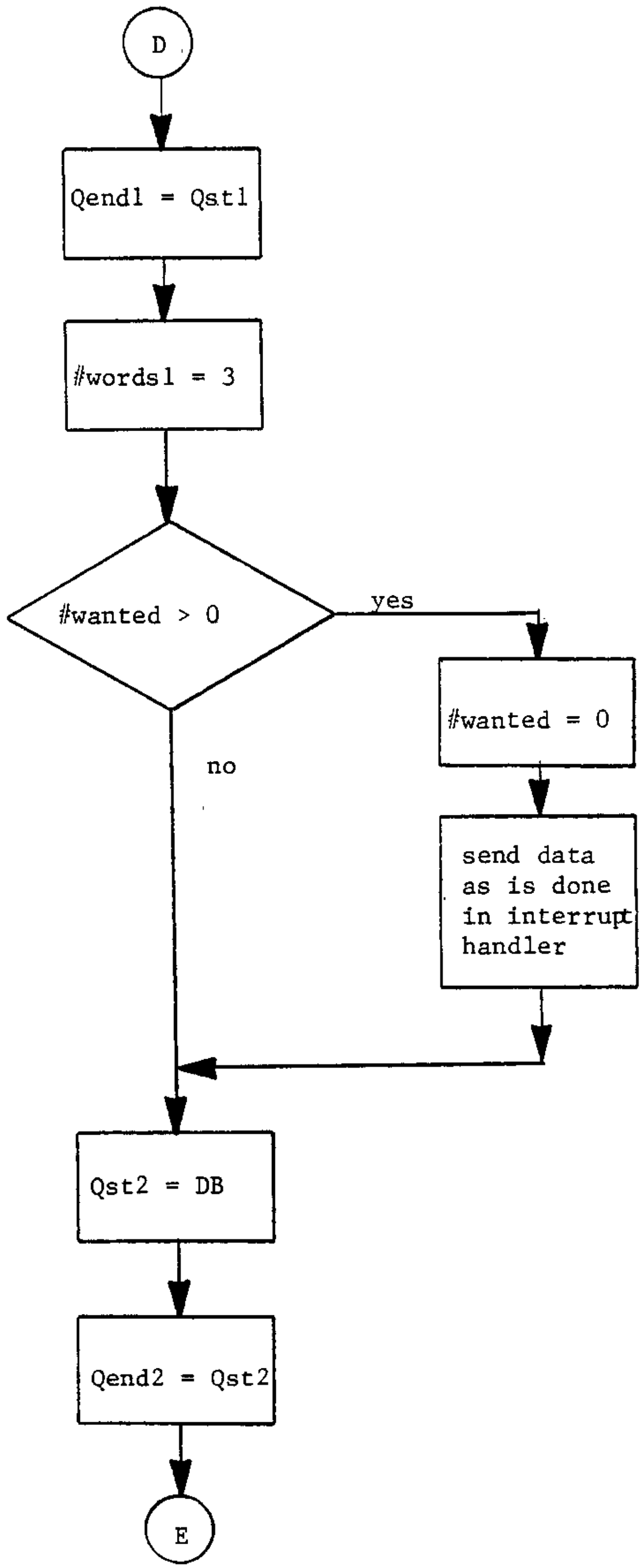
The update task's job is to read data from the NAV computer, the control computer, the pressure sensor, and the compass and update the command computer data base. Because of the specialized nature of these I/O operations modified read and write routines must be used. The write routines must allow overwriting and the task can not wait for reads to be completed. In addition to this, when Z, Z dot, and compass are being written, tasks and interrupts should be blocked so the information is transferred as a unit. Given these constraints and the channel buffer definitions the following flowchart illustrates the details of the update task.

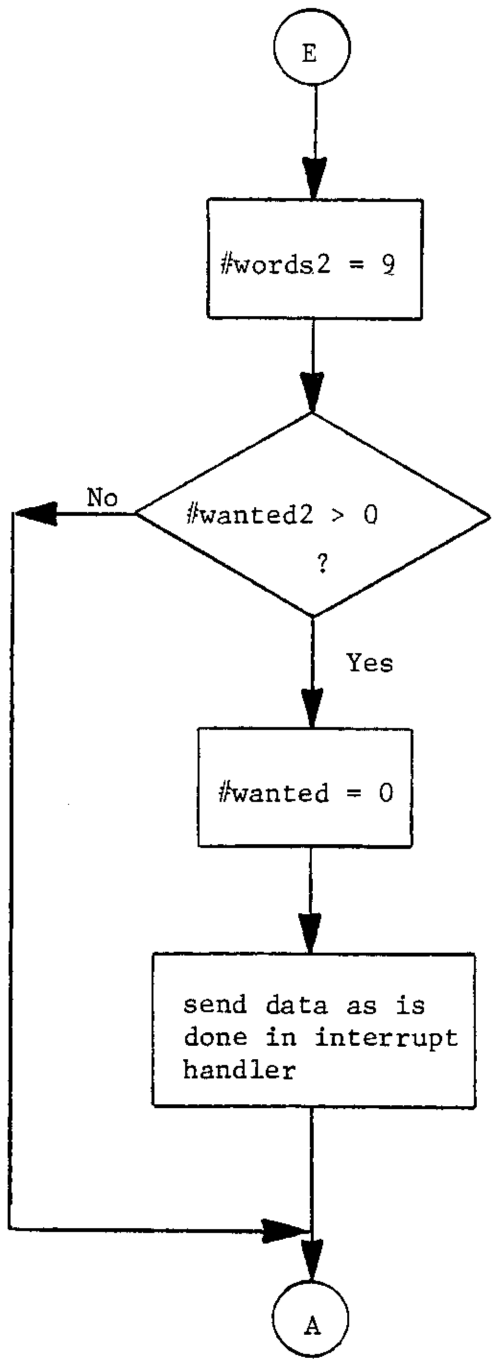
COMMAND COMPUTER DATA BASE UPDATE TASK




B
Buffer Ready Flagl
No
Yes
Clear Flag
Block Tasks
Send a request over channel 1 for 7 words
wait for proceed signal
unblock tasks
C


C
read pressure transducer
calculate Z value
calculate Z dot
read compass
block tasks and interrupts
load data base words 8-10 with Z,Z,compass
Qstl = DB+7
D

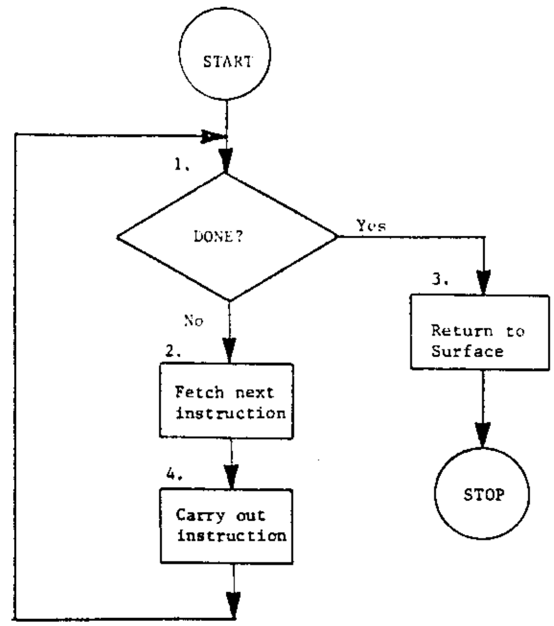

D
Qend1 = Qst1
#words1 = 3
#wanted > 0
yes
no
#wanted = 0
send data
as is done
in interrupt
handler
Qst2 = DB
Qend2 = Qst2
E


E
#words2 = 9
No
#wanted2 > 0
?
Yes
#wanted = 0
send data as is
done in interrupt
handler
A

5.1.3 MISSION TASK/MISSION CONTROLLED TASKS

This mission task is the task in the command computer which executes a list of preprogrammed commands. The commands are of the same format and type as commands issued from a terminal. To issue a command, the mission task should block other tasks and then write the command to follow flag followed by the desired command just as though it were typed from a terminal. Once the mission has been completed the vehicle is instructed to swim to the surface.

The mission controlled tasks, which are variable in number, may be controlled directly by the mission task using the W and C commands. This control allows the scientist to not only control the vehicles movement, but also allows for the control of peripheral equipment such as video, acoustic, and tactile devices. It should be noted that these auxillary tasks need not be controlled exclusively by the mission task and may be controlled through manual operator control.



1. DONE?

Checks a flag which is set true by 4. (Carry out instruction) upon execution of the FINISH instruction.

2. Fetch next instruction

Loads the instruction and parameters and increments the command pointer (CP).

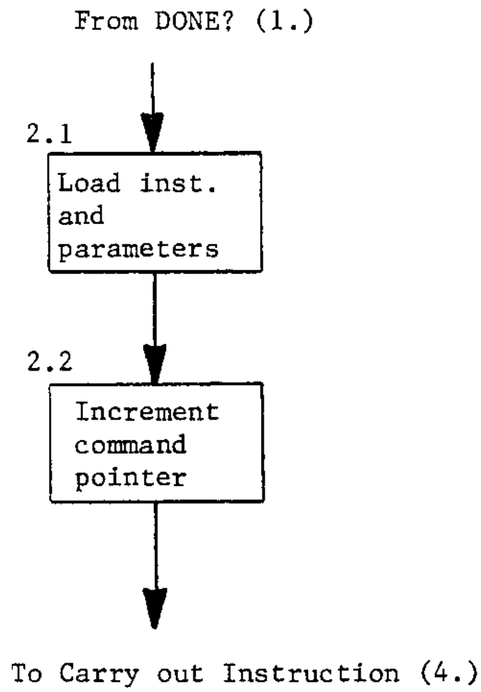
3. Return to surface

Causes the vehicle to return to the surface for recovery. It sends the control the instructions and parameters to travel to the surface (MOVEZ 000).

4. Carry out instruction

Checks to see if the instruction is for the control computer or an instruction to be carried out by the command computer. If the instruction is for the control computer, it is sent the appropriate information. If the instruction is for the command computer, the correct operations are executed. After initiating the operation the task waits until the command finished flag is set true.

2. Fetch next instruction



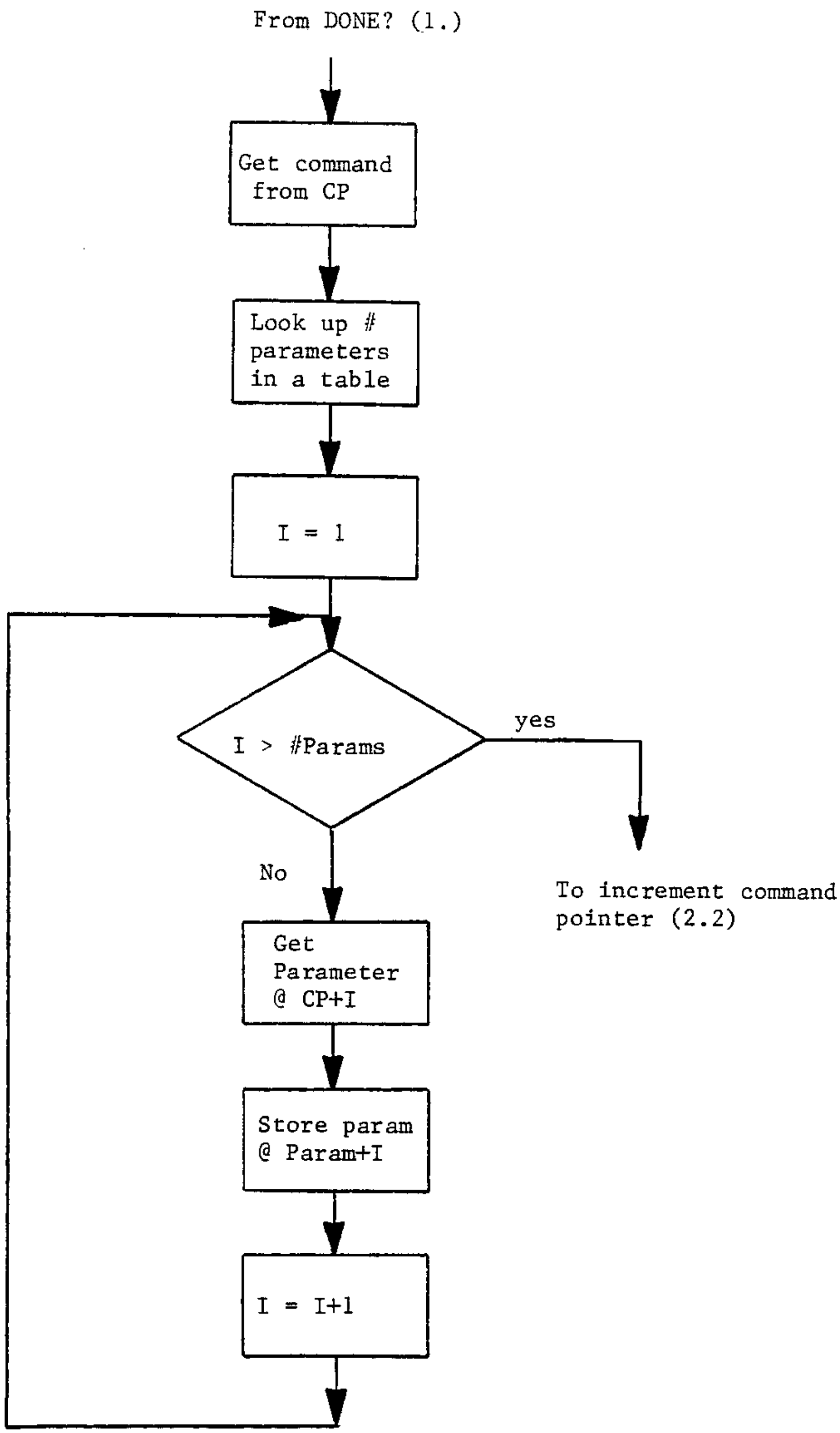
2.1 Load instruction and parameters

The instruction is read from the command list @ CP. The parameter(s) are then read from the command list starting @ CP+1. The number of parameters read depends on the instruction read. The available instructions are the standard command scanner commands.

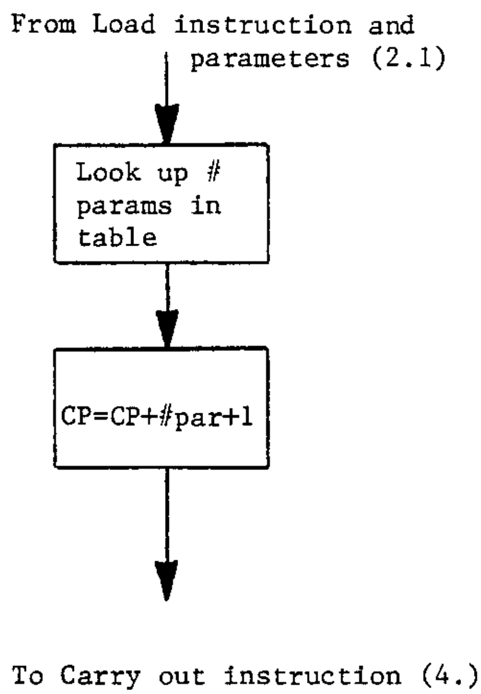
2.2 Increment command pointer

The command pointer is incremented by the number of words required by the current instruction. If the CP is out of bounds then it will be set to a recovery command.

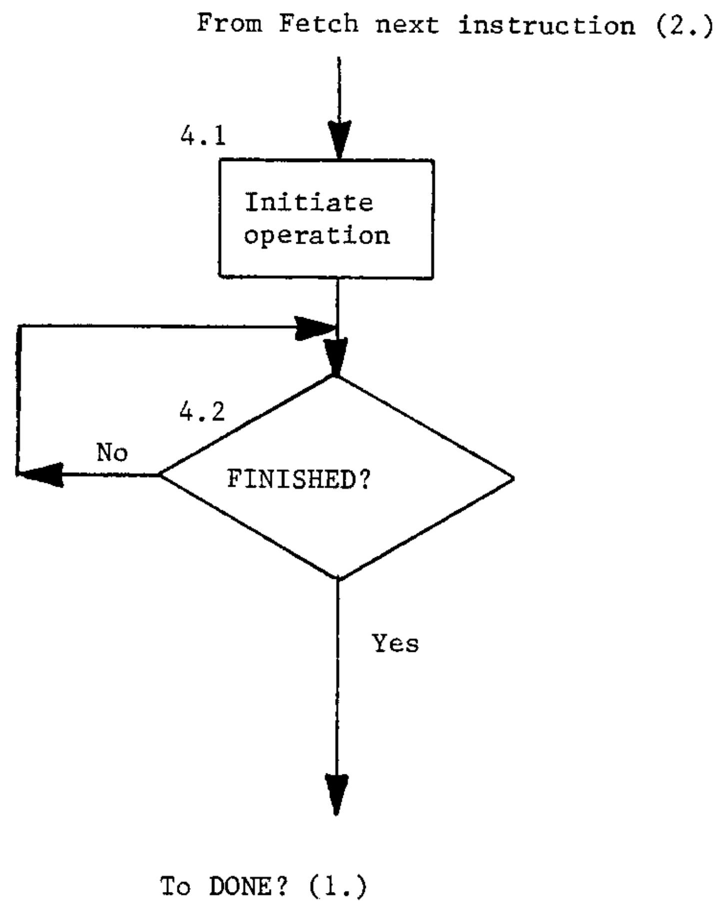
2.1 Load instruction and parameters



2.2 Increment command pointer



4. Carry out instruction



4.1 Initiate operation

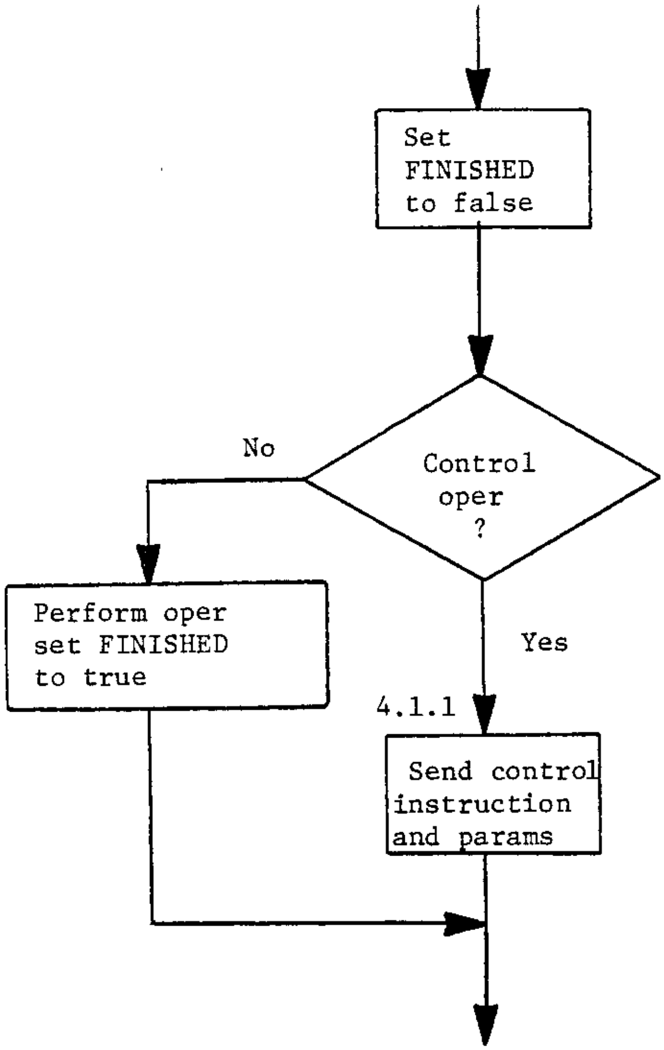
If the operation involves vehicle maneuvers, send the control computer the operation to be performed and the associated parameters. If the operation does not involve control, execute the command (probably I/O) to carry it out. When the command list is exhausted, DONE is set true.

4.2 FINISHED?

A wait loop where the task sits until it receives a signal that the operation is complete.

4.1 Initiate operation

From Fetch next instruction (2.)



To FINISHED? (4.2)

4.1.1 Send control instruction and parameters

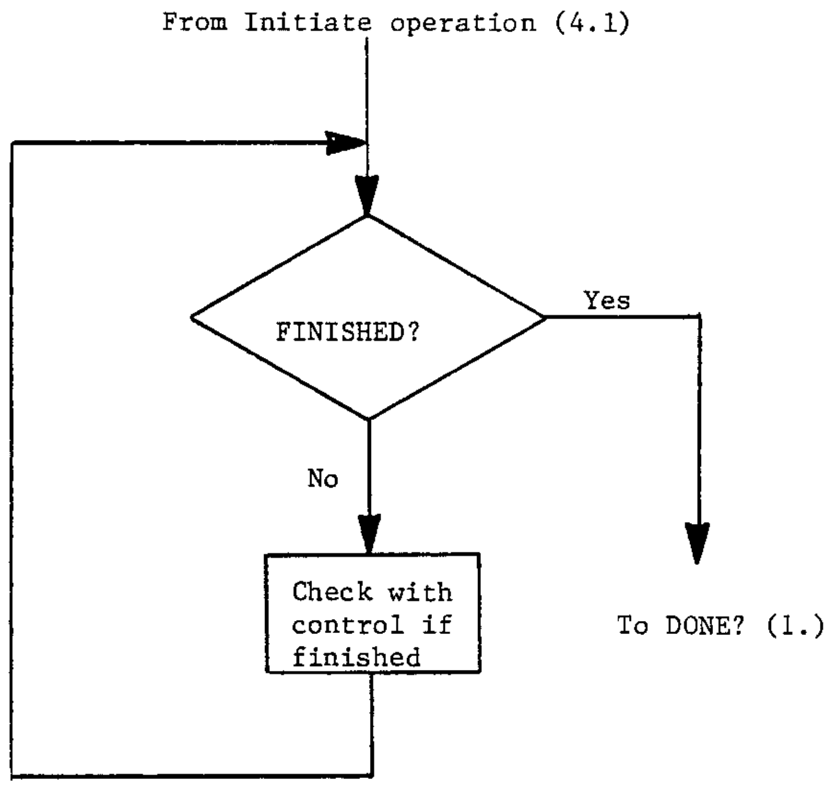
Send the command to the control computer via the command scanner. The general operations taking place are:

send command to follow flag

send parameters separated by commas

send command letter

4.2 FINISHED?


From Initiate operation (4.1)
FINISHED?
Yes
No
Check with control if finished
To DONE? (1.)

5.2 CONTROL COMPUTER

The control computer directly controls the vehicles thrusters and thus its movement. It receives its commands via a standard command scanner. There are two types of vehicle control, manual operator control and automatic navigation. Manual operator control is used to control the thrusters directly ignoring any position/orientation information. Automatic navigation makes full or partial use of the position and orientation to maintain a particular altitude and heading, or to travel from place to place in a straight line. Within the control computer there are two tasks, the Auto Altitute task and the Travel task. Given a Z value the Auto altitude task will keep the vehicle at that altitude until the task is pended or a new Z is received. The Travel task accepts X,Y,Z,O and maneuvers the vehicle in a straight line to the Z altitude and then to the X,Y, and then turns to heading O. Once in position, the Travel task maintains the vehicles position until the task is pended or a new X,Y,Z,O is received.

Manual operator control allows access to individual thrusters. Depending on which manual command is given, the Auto altitude and/or Travel tasks may be pended. When not pended, both the Auto altitude task and the Travel task access a data base (Figure 11) which contains destination, position/orientation, an thruster speed information. The data base is also used as an input/output buffer as is done in the command computer.

Figure 11

CONTROL COMPUTER DATA·BASE

DB			
	1	reliability word	
	2	rel head	X
	3	range	Y
	4	rel head dot	abs head
	5	range dot	X dot
	6	---	Y dot
	7	---	head dot
	8	Z	
	9	Z dot	
	10	compass	
	11	TX	
	12	TY	
	13	TZ	
	14	TO	
	15	new position flag	
	16	newX	
	17	newY	
	18	newZ	
	19	newO	

5.2.1 CONTROL COMPUTER I/O CHANNEL DEFINITIONS

A. Control computer - Command computer I/O channel 2 used by Auto altitude and Travel tasks in control and update tasks in command

in control computer the output buffer word are:

Buffst2	points to word eleven in the control computer data base
Buffend2	points to the fifteenth word in the data base (14th + 1)
Qst2	will almost always point to the eleventh word in the data base because any time the Auto altitude or Travel tasks are running, the information is updated continuously.
Qend2	will almost always point to the eleventh word because the buffer should almost always be full.
#words2	will almost always be four because the buffer is almost always full

#wanted2 will be either four or zero most of the time it will be four because the command computer re-issues its read request as soon as a read is done.

UART addr holds the five bit address of the UART linking the command computer and the control computer.

in control computer th input buffer words are:

Read buffer pointer2 points to word one of the control computer data base.

Buffer ready flag2 usually indicates not ready because the command computer reads the data as soon as it is ready.

UART addr holds the five bit address of the UART linking the control computer and the command computer.

B. Control computer - Command computer I/O channel 3 used by Travel task and Mission task

in control computer the output buffer area words are:

Buffst3 points to the Done flag in the Travel task

Buffend3 points to the location following the Done flag

Qst3 will always point to the Done flag

Qend3 will almost always point to the Done flag because the buffer will usually be empty and that is the first free location in the buffer.

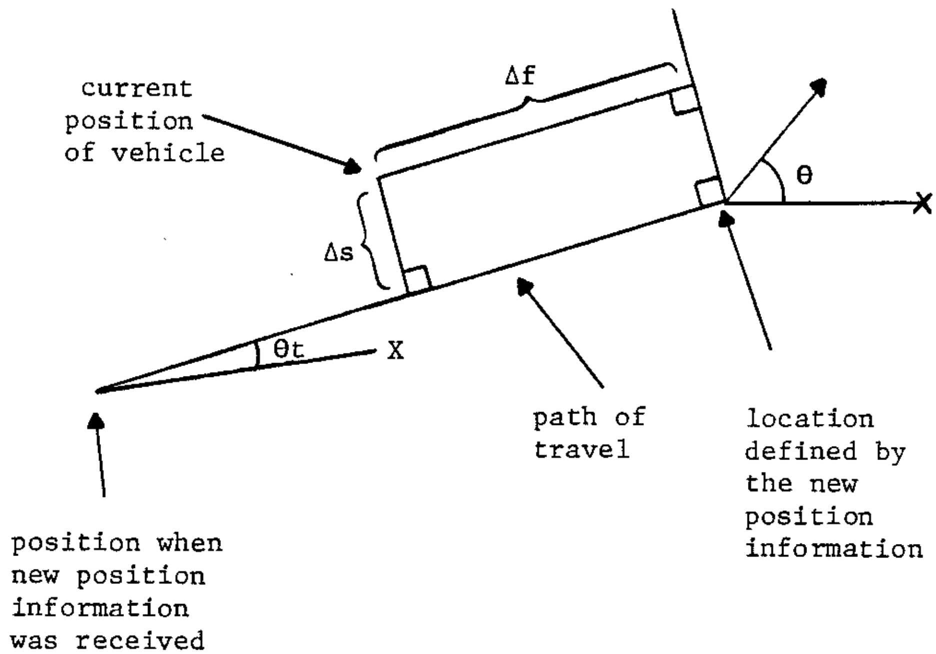
#words will almost always be zero because the command computer reads the done flag almost immediately.

#wanted will almost always be one because the command computer is usually waiting for the control computer to finish a command.

UART addr holds the five bit address of the UART linking the command computer and the control computer.

5.2.2 CONTROL COMPUTER TRAVEL TASK

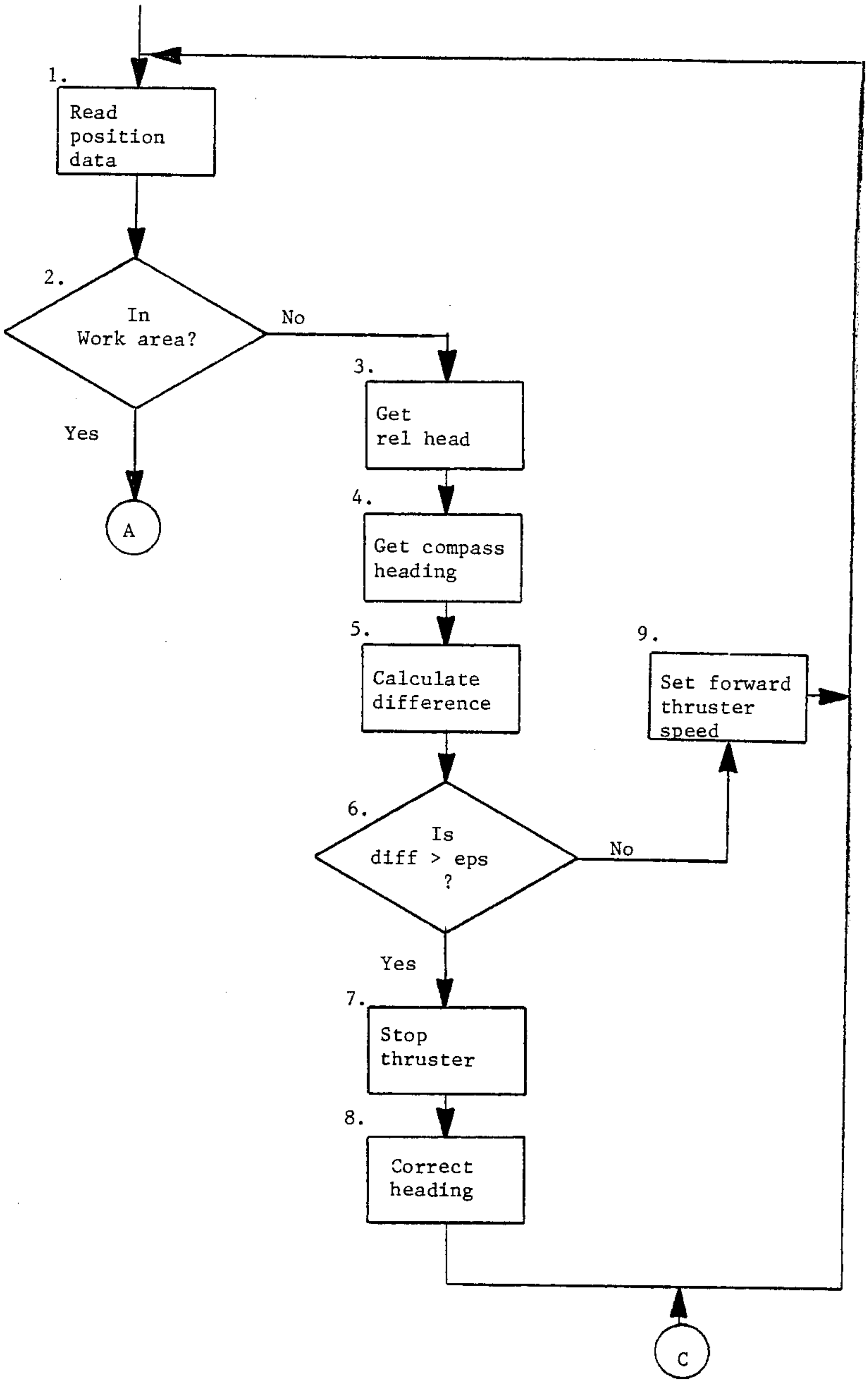
The Travel task accesses a common area which describes the desired position and orientation of the vehicle and maneuvers into that position. In general, the task reads the position data and checks to see if the vehicle is in the work area. If outside the work area the vehicle is maneuvered toward the transponder being used by the NAV computer to get the heading and range values. The vehicle is moved toward the transponder until it is in the work area. Once in the work area the task checks to see if a new position/orientation description has been receivedd since the last time through the control loop. If new data is present then two pairs of constants and a heading for the path of travel (Θt) is calculated. The path of travel is defined to be the line in the X-Y plane connecting the position of the vehicle at the present and the new position described. The two pairs of constants which are referred to as Δs and Δf constants are used to calculate the perpendicular distance from the vehicle to the path and to the destination endpoint of the path. The distances are signed values referred to as Δs and Δf and are used to set sideslip and forward thruster speeds.

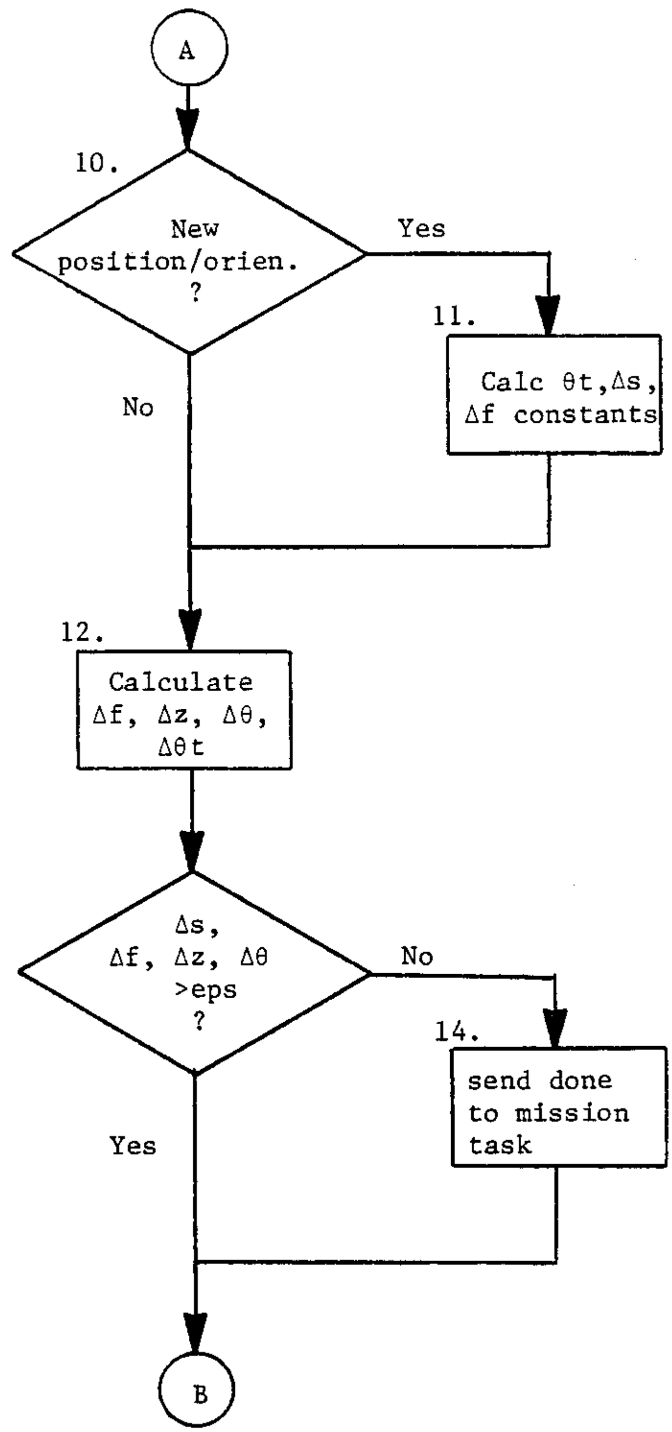


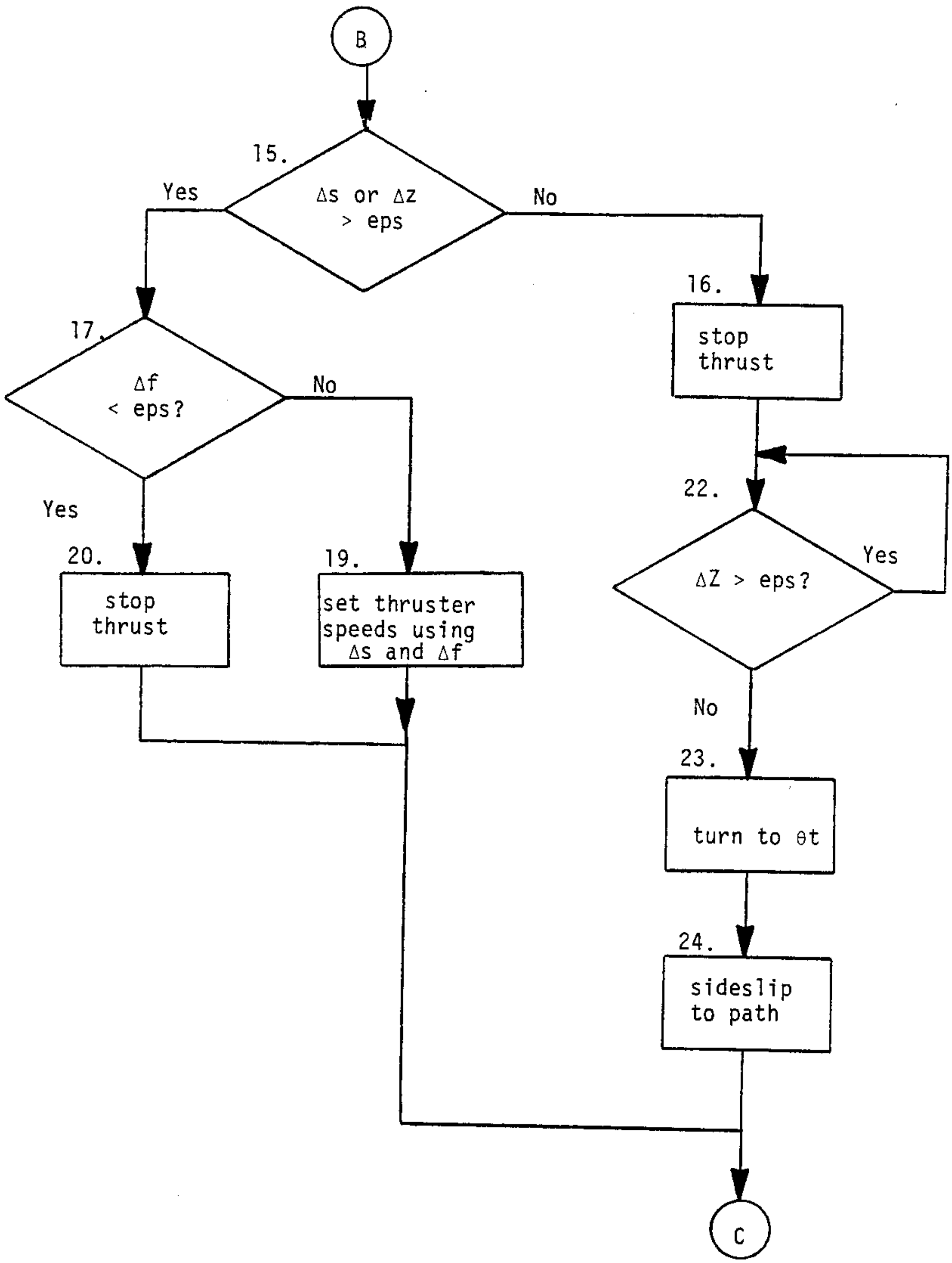
Following the new position check or constant calculation and Δs and Δf values are calculated using the constants. The ΔZ, which is the difference between the current alt and the desired alt, is calculated as well as $\Delta\Theta t$, $\Delta\Theta$. $\Delta\Theta t$ is the difference between the current heading and the heading of the path of travel. Θ is the difference between the current heading and the desired heading defined by the new position information. If Δs, Δf, Δz and $\Delta\Theta$ are all small then the task will send a command complete signal to the command computer. In addition to this, the Δs and Δf constants are recalculated such that they refer to a path through the destination in the same direction as the desired heading. Θt is also set equal to Θ.

If either Δs or ΔZ are too large, all thrusters are stopped and the vehicle is moved up the path as directly as possible. If Δs and ΔZ are both small then Δf is checked, when Δf is small, the vehicle is rotated from θt to θ which is the desired position. When Δf is large, the forward and sideslip thrusters are set using the Δf and Δs values. This control loop is continued until the travel task is pended or a new position is received.

Control computer travel task




A
10.
New position/orien. ?
Yes
11.
Calc θt,Δs, Δf constants
No
12.
Calculate Δf, Δz, Δθ, Δθt
Δs, Δf, Δz, Δθ >eps ?
No
14.
send done to mission task
Yes
B



B
15.
Δs or Δz > eps
Yes
No
16.
stop thrust
17.
Δf < eps?
No
Yes
20.
stop thrust
19.
set thruster speeds using Δs and Δf
22.
ΔZ > eps?
Yes
No
23.
turn to θt
24.
sideslip to path
C

Travel Task -- Flowchart Explanation

1. Read position data

Perform a standard read over channel 2 for ten words.

2. In work area?

Check bit 10 of the reliability word; if it is set, then the vehicle is in the work area; if not set, it is outside the work area.

3. Get rel. lead

Get the relative heading from word two of the control computer data base.

4. Get compass heading

Get the compass heading from word ten of the data base.

5. Calculate difference

Subtract one from the other and take the absolute value of the difference.

6. <u>Is difference $\geq$ eps?</u>

Check to make sure the vehicle is on course toward the work area.

7. <u>Stop thrusters</u>

The vehicle is off course. Stop all thrusters for corrective measures.

8. <u>Correct heading</u>

Check the sign of the difference and turn in the proper direction, until the heading is correct.

9. <u>Set forward thruster speed</u>

Set the forward thruster speed as a function of the range. If at a greater distance, go faster. T_x = Range * k_x.

10. <u>New position/orientation?</u>

Check word fifteen of the data base, if it is set (non-zero) then new data is present.

11. Calculate Θt, Δs, Δf constants

Δs and Δf discussed earlier may be calculated using the following equations:

$$\Delta s = X_s * X + Y_s * Y + B_s$$

$$\Delta f = X_f * X + Y_f * Y + B_f$$

the constants X_s, Y_s, B_s, X_f, Y_f, B_f may be calculated as follows:

Given $X_\emptyset$, $Y_\emptyset$ is the current position

and X_1, Y_1 is the destination position,

$$M_s = (Y_1 - Y_\emptyset)/(X_1 - X_\emptyset)$$

$$B_s = Y_1 - M_s * X_1$$

$$X_s = \text{square root } (M_s/M_s^2) + 1$$

$$Y_s = \text{square root } (-1/M_s^2) + 1$$

$$X_2 = Y_1 - (X_\emptyset - X_1)$$

$$Y_2 = X_1 - (Y_\emptyset - Y_1)$$

$M_f = (Y_1 - Y_2)/(X_1 - X_2)$

$B_f = Y_1 - M_f * X_1$

X_f = square root $(M_f/M_f^2) + 1$

Y_f = square root $(-1/M_f^2) + 1$

12. Calculate Δs, Δf, Δz, $\Delta\Theta$, $\Delta\Theta t$

Calculate the difference between the vehicle's current position and orientation and what it should be according to the description in the data base.

$\Delta s = X_s * X + Y_s * Y + B_s$

$\Delta f = X_f * X + Y_f * Y + B_f$

$\Delta z = [z - zb]$

$\Delta\Theta = [\Theta - \Theta db]$

$\Delta\Theta t = [\Theta - \Theta t]$

where X, Y, Z, Θ are the current position and heading.

13. Δs, Δf, Δz, $\Delta\Theta \geq$ eps

If all of these are less than eps then the vehicle is in a satisfactory position.

14. Send done to mission task

Send the one word done flag over channel three using a write routine which allows overwriting.

15. Δs or $\Delta z \geq$ eps

if Δs or Δz are too big, the vehicle is badly off course.

16. Stop thrust

If vehicle is off course, stop thrusters and take corrective measures.

17. $|\Delta f| \geq$ eps

If Δf is small enough then the vehicle is at the destination.

18. Stop thrust

Turn off thruster for the turn to the desired heading.

19. <u>Set thruster speeds using Δs and Δf</u>

Set the thruster speeds using Δs and Δf

$$T_x = f * k_x$$

$$T_y = s * k_y$$

20. <u>Turn to θ</u>

Turn vehicle to heading θ, the final orientation.

21. <u>Recalculate Δs and Δf constants, set $\theta t = \theta$</u>

Calculate Δs and Δf constants in the same fashion as in 11 using the current position and a point on the ray eminating from the current position at heading θ.

$$X_1 = X_0 + 10 * \cos\theta$$

$$Y_1 = Y_0 + 10 * \sin\theta$$

22. <u>$|\Delta Z| \geq$ eps?</u>

Calculate Δz and check if it is > eps. Keep looping until a small enough Δz is found.

23. <u>Turn to θt</u>

Correct the heading for path correction within eps/2.

24. <u>Sideslip to path</u>

Sideslip until Δs becomes eps/2, then proceed.

5.3 COMMUNICATION COMPUTER

The communication computer will act as a "fancy UART" between the command computer and a terminal on the surface. Given this approach it can essentially take the place of the TTY connected to the command computer once the vehicle is untethered. Being treated as a TTY, the communication computer should use the same I/O protocol as a TTY.

5.4 NAVIGATION COMPUTER

The navigation computer is somewhat unique in that it is a well-defined subsystem of the SIMS vehicle. A complete description of this subsystem is included in section 6.0 of this report.

5.5 AN INTERPROCESSOR I/O SCHEME

The proposed solution to the interprocessor, intertask communication problem was formulated around the idea that there exists a need for reliable, standardized I/O interfaces. The primary concern of this scheme is the avoidance of deadlock situation. When tasks in two processors are communicating over a single serial line, it is very easy for one task to tie up, other tasks start waiting to use the port and eventually each task will be waiting and none will be executing. From this you should see that it is important that any data written to the UART should be read immediately.

Given these concerns the I/O scheme may be summarized as follows:

5.5.1 INTERPROCESSOR WRITING

Each task within each processor will have its own channel write routine and an output buffer area for each task to task channel. To access the write routine the accumulator is loaded with the data to be sent and a second parameter-passing variable will hold the ID of the channel. Using these, the outgoing data may be buffered until it is needed. The buffer itself is a ring buffer of a minimum number of locations. The buffer must be able to hold as many words as could be requested in one request. When the buffer becomes full due to more writes than reads, the write routine just waits until data is read, opening up space in the buffer. After putting data in the buffer, the write routine checks to see if there are any unsatisfied read requests. If there is a request outstanding, the write routine sends the next block of data in the buffer.

The global theme of interprocessor writing is to buffer everything in the task where it is written and to only send data through the UART when you have a known request for it. Given this, the UART will never get tied up by a single task.

5.5.2 <u>INTERPROCESSOR READING</u>

Each task within each processor will have its own channel read routine and an input buffer area for each task to task channel. To access the read routine, the accumulator is loaded with the number of words being requested (1-16) and a second parameter-passing variable will hold the ID of the channel. Data which is read, is loaded into memory starting at the location pointed to by the read buffer pointer in the input channel area. Since no data is transferred unless a request is made for it, the read routine sends the other processor a request for data from a particular task. After making the request the routine waits in two wait loops until the desired data is received. Control is then transferred back to the calling routine.

5.5.3 INCOMING INTERRUPT HANDLER

The interrupt handler is responsible for all input to the processor from other processors. There are four types of data coming into the processor, read requests, write data, data-not-available flags, and command-to-follow flags. On receiving an interrupt, the ID of the channel is saved and the type of the interrupt, one of the four mentioned, is determined.

When a read request is made, the interrupt handler makes a check to see if the number of words requested is available. If the data is available then it is sent immediately, otherwise a data-not-available flag is sent and the output buffer is flagged as having an outstanding request.

When a write data signal is received, the interrupt handler reads the data from the UART, two characters each containing a six bit byte of the twelve bit word. The data read is stored starting at the location pointed to be the read buffer pointer.

When a data-not-available flag is received, it is simply loaded into the buffer read flag for that command.

When a command-to-follow flag is received, control is simply transferred to the command scanner with the address of the UART passed in the accumulator.

5.5.4 <u>INTERRUPT PREPROCESSING</u>

All UART interrupts, TTY and processor, call a routine which sets up two routines to access that particular UART. These routines are used by the interrupt handler and the command scanner to do further reading and writing to and from the UART. The routines are simply a short read routine and a short write routine into which are inserted the proper read, write, and skip commands to access the interrupting UART.

5.5.5 CHANNEL BUFFER I/O TABLE

When doing I/O over different channels the interrupt routine and read and write routines must be able to access the appropriate buffers. Because of this need, a table of channel ID's and buffer addresses is set up to provide the correct association. The table below illustrates a possible ID-address scheme.

Channel ID	Input Buffer Area Address	Output Buffer Area Address
1	IN BUFF1	OUT BUFF1
2	IN BUFF2	OUT BUFF2
3		

2 bits	4 bits	2 bits
type	# words	channel ID

Channel ID

refers to the I/O channel on which the I/O is taking place

type

00 - read request
11 - write data
10 - data not available
01 - command to follow

words

words ranges from 1-16 and in the case of read requests and write data signals it refers to the number of words being requested or sent

unused	high order 6 bits
unused	low order 6 bits

all actual data is transferred in two eight bit words, six bits in each word. The data is packed with the six bits packed in the low end of the word.

5.5.6 LIST OF COMMANDS (RECOGNIZED BY THE STANDARD COMMON SCANNER)

O/S and editing commands

Command	# of params	Explanation
.	1	change fields
cr	Ø	close current memory location
/	Ø/1	open the current memory location
lf	Ø	close the current memory location and open the next
W	1	pend task (wait)
C	1	unpend task (continue)
O	Ø	turn on clock
X	Ø	turn off clock
K	Ø	abort mission
Q	Ø	query
G	1	start execution in location
,	2	used to separate parameters (nn,nn)
ctl-U	Ø	ignore any partially input command
I	Ø	initialize processor, I/O channels, etc.

Thruster commands

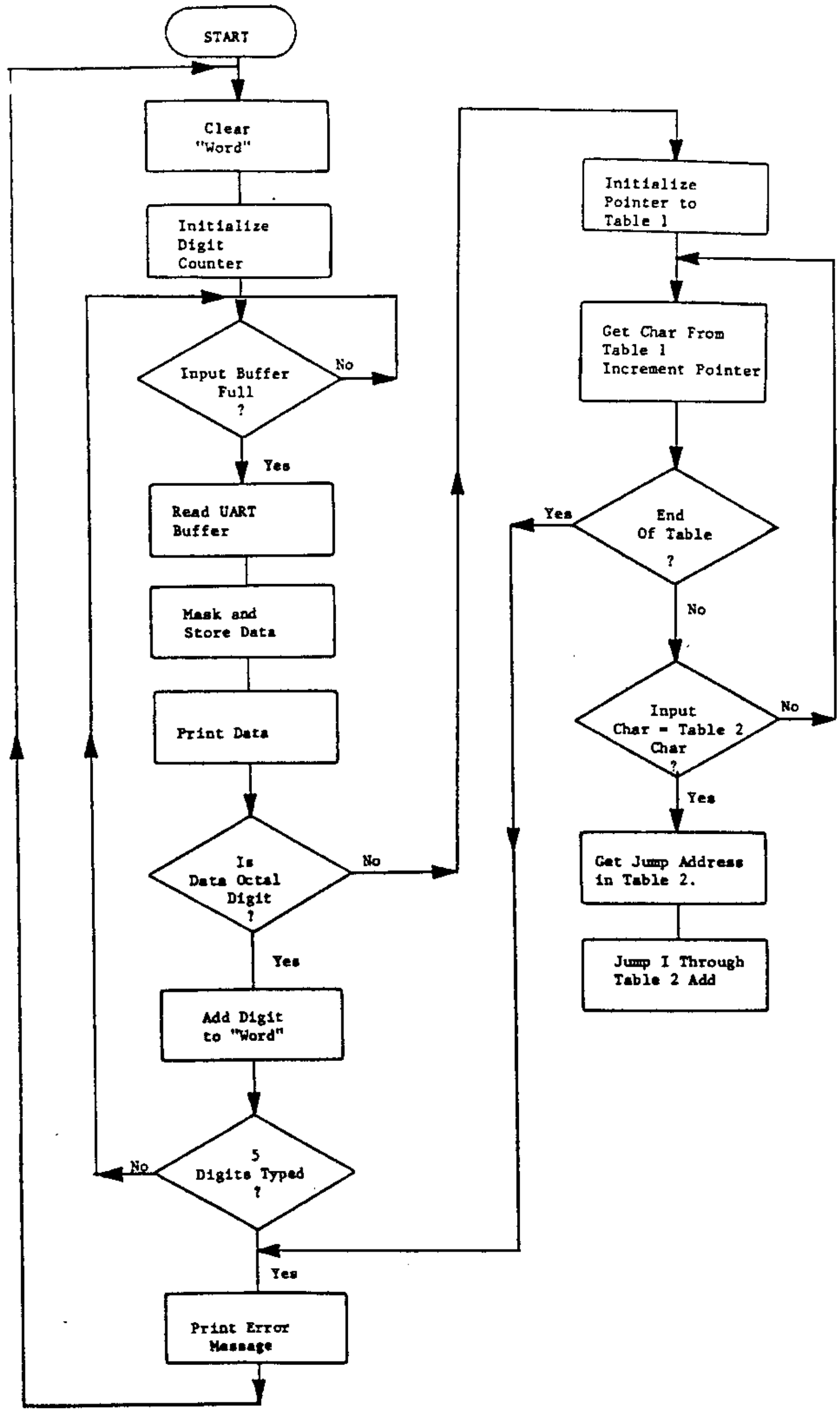
Commands	# of Params	Explanation
U	1	set UP thruster speeds
D	1	set DOWN thruster speeds
F	1	set FORWARD thruster speeds
B	1	set BACK thruster speeds
L	1	set ROTATE-LEFT thruster speeds
R	1	set ROTATE-RIGHT thruster speeds
S	1	set SLIP-STARBOARD thruster speeds
P	1	set SLIP-PORT thruster speeds
H	Ø	halt all thrusters
A	1	move to altitude
T	1	turn to heading
M	3	move to position X,Y,Z
+	2	thruster forward at speed
-	2	thruster back at speed
N	Ø	continue the current move command

General command format

```
nnnn, nnnn, ... nnnn L
```

where nnnn is a parameter to the command. The commas are used to separate one parameter from the next. L is the single letter command.

COMMAND SCANNER



Interpretation of commands by command computer

K - pend mission task, send control a command to swim to the surface.

Q - causes a jump to a subroutine which dumps information to a UART.

I - initialize RAM locations, command communication, control and NAV to do the same initialization. Start the update task.

U,D,F,B,L,R,S,P,H,-,+,A,T,M,N - all these commands are simply relayed to the control computer.

Interpretation of commands by NAV computer

K - stop execution of the NAV task

Q - dump information in the same fashion as command computer

I - initialize RAM locations, start up the NAV task for communication and position information.

--- thruster commands are ignored

Interpretation of commands by the control computer

U,D,H - pend Auto alt. task and Travel task, turn off thrusters, then set thruster speeds.

F,B,L,R,S,P - pend Travel task, turn off all horizontal thrusters, then set thruster speeds.

A - pend Travel task, turn off thrusters, then set the new altitude.

T - if running Travel task just change (-) in common and set new data flag. If not running Travel task, start up with current position + alt.

M - set Auto alt. and Travel tasks with the requested values.

N - restart the Travel task and Auto alt. task without giving them new values. Used primarily for testing purposes.

+, - pend Auto alt. and Travel tasks stop all thrusters and set thruster speed.

K - pend all tasks and set up thrust slow

Q - dump information

I - initialize RAM locations, Auto alt. and Travel tasks.

6.0 NAVIGATION WITHIN A STRUCTURE

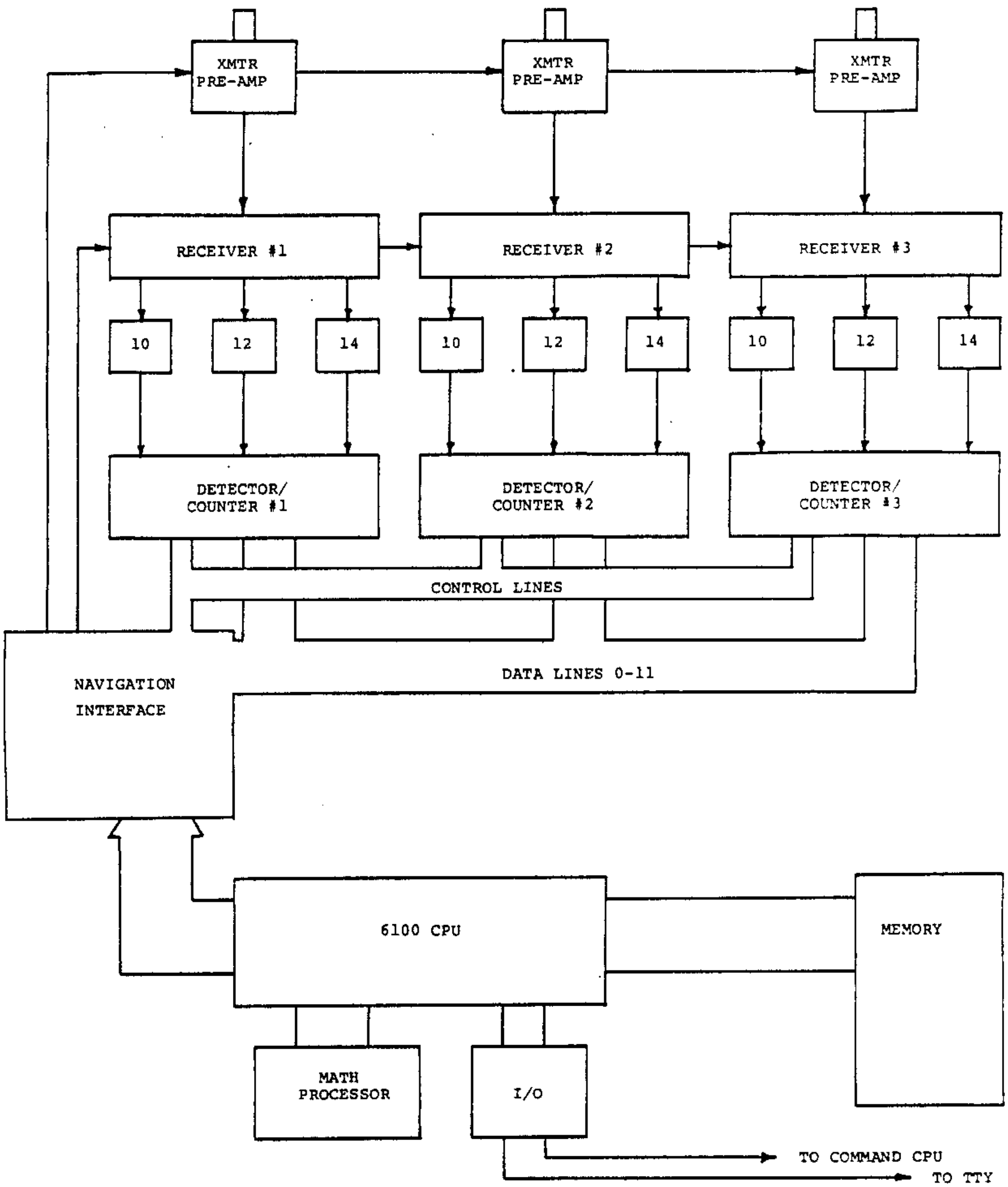
6.1 INTRODUCTION

The objective of the navigation problem is to provide to the main vehicle control system, the positional information necessary to navigate the vehicle to an underwater structure, and move through or around the structure and return.

The vehicle will be initially placed or submerged somewhere between two to three hundred feet from the target area. The vehicle is equipped with a 3-dimensional numerical map (structure coordinate system) stored in memory containing the structure coordinates of the target, and the positional coordinates of three object transponders (whose purpose is analogous to that of airport beacons). The depth or z coordinate of the vehicle is provided by the main system computer, through a depth sensor.

Figure 12

SIMS NAVIGATION SYSTEM



6.2 SYSTEM DESCRIPTION

The vehicle is equipped with three transducers mounted on an equilateral triangle of three foot length. Each transducer is connected to one transmitter and one receiver with three detector counters. Using one of the three transmitters, a pulse of 95 KHz sonar is transmitted to the three object transponders in the water. When the object transponders sense the transmitted frequency, each, after a fixed delay, returns a pulse, which is detected by three of the nine counters contained on the vehicle. The object transponders each transmit an acoustic pulse of a frequency between 110 and 130 Khz (resettable in 2 Khz steps). One detector at each transducer on the vehicle is set to sense a specific frequency. When the signal is detected the corresponding counter, which was started at the initial transmission, is stopped. From one 95 Khz interrogation pulse, nine two-way travel times may be obtained. From these numbers, the navigation computer must provide the necessary information.

6.3 NAVIGATION COMPUTER TASK DESCRIPTION

During a ping, the counters measure the length of time for the sonar signal to travel from the transmitter on the vehicle (pinger), to the object transponders in the water, the turn-around time, and the return time to the vehicle receivers. For convenience and clarity the term "hydrophone" will be used to designate the vehicle receivers and transmitters. Knowing the speed of sound through the water, the distance from each transponder can easily be computed.

Using the coordinates of the object transponders and the respective distances from them, a position in an x-y plane can be computed as well as an absolute heading (angle in x-y plane between the vehicle longitudinal axis and a line connecting the vehicle with one of the object transponders). When the vehicle is outside of a two hundred foot range, the navigation computer must provide a relative heading, a range (from one of the object transponders), and their derivatives (relative heading dot and range dot). When the vehicle is within the two hundred foot range the navigation computer is responsible for vehicle x-y coordinates, absolute heading and their derivatives (x velocity, y velocity, heading dot).

6.4 DISCUSSION

The scenario as so far stated is idealistic. There are physical problems that exist when using sonar in water. There are problems such as multipath where a signal bounces off the surface or the floor, thus increasing the distance of the sonar travel time. Shadowing where an object blocks the signal entirely, as well as sound velocity in water errors (this varies due to change in temperature, density, salinity, etc.), and errors in placement of the object transponders are all problems which could produce incorrect data. Therefore, before any calculations can be done, the corrected raw data (corrected from two way travel time) must be checked for validity. This is done by comparing the raw data with a window, which is a predicted value based on history. By adjusting the width or size (tolerance) of the window a small range of variance can be provided allowing for small errors and displacement from real time.

Initialization of the windows at the beginning of operation is done by obtaining a number of returns and checking for uniformity. When enough data is consistent, a window is formed using the most recent sample. At a large range all transponders may not be heard. In such a case, only windows for data received are to be initially set. Windows not set are initialized directly on the fly as the returns begin to be received.

Since the windows must change as the vehicle moves, they must be continually updated, and since the windows are based primarily upon the previously received sample then there must be a way to determine if a window has been set correctly. In order to detect a faulty window, a coded matrix was devised. The contents of the coded matrix is based upon the element by element comparison between the raw data in matrix form and its corresponding elements of the windows in matrix form, and window sizes in matrix form. If a raw data term is within its window and window size then it is termed as a "good return". If the return is outside of the window and window size, it is termed a "bad return". An element for which no return was obtained is termed a "no return". If a certain element is consistently a bad return then the window may be in error. If this occurs, then the window size is expanded slightly. If the returns continue to be bad, then this window size is expanded slightly. If the returns continue to be bad, then this window is completely discarded, and intialized on the fly. This type of operation relies on an assumption that the samples will be more often accurate than not.

Beyond the use of the coded matrix towards window adjustment is the ability to determine what information can be computed. The mathematics used to compute position uses the horizontal distance from two object transponders, which yields two solutions mirrored about a line drawn through the two transponders. To disambiguate, either history (previous samples) or the distance to the third transponder must be used. If the distance to the third unused transponder is available, then by computing that distance twice using the two solutions, it can be determined which solution is closest. If history is available, then by comparing the two solutions with history, the closest one can be determined. If history is available, then upon examination of the coded matrix if there are at least two "good returns", a position can be calculated. If no history is available then three good returns must be received. A relative heading is computed using the distance from one object transponder to each of the three hydrophones. By examining the coded matrix it can be decided if enough good returns have been received to compute the relative heading. Note that if the coded matrix is arranged as in Figure 13, the position check is made with the rows, and the relative heading check is made with the columns. Also, the absolute heading must have the same information as the relative heading as well as the information necessary to compute position.

A decision as to which hydrophone to ping is made using the coded matrix. By checking the rows of the matrix, it can be determined which row (hydrophone) has more good returns. For consistency one pinger must be used over the others. This is done as a priority tie breaker.

Figure 13

CONFIGURATION FOR RAW DATA,
WINDOW, AND WINDOW SIZE MATRICES

Object Transponders

Vehicle Hydrophones

	A	B	C
1			
2			
3			

If all the returns in a column are consistently bad or no return status even after window correction, then failure of either the counters or the transponder is indicated. If the same conditions exist for a row, then a failure of the hydrophone is indicated. For each time one of these conditions occur in the coded matrix a hardware error is recorded. A running count of the hardware errors is kept, if the number of errors over the number of samples is large enough, then a malfunction condition exists. When the vehicle is within a one hundred foot range then the probability of shadowing due to the structure increases and any error analysis is invalid. Also outside of the two hundred foot range, one or more of the object transponders may be out of range of reception in which error analysis is again invalid. The coded matrix allows window correction, "one-look" information availability, pinger decision, and elementary error analysis.

The differential calculations are made using the difference between present (last sample) information and information stored in history over one cycle time. Due to memory limitations only one set of samples (group of information generated by one pass of the algorithm) can be stored in history.

This is the theory upon which the algorithm operates. Following is a step by step description of the algorithm from which one may gain a more concise image of the navigation.

6.5 NAVIGATION ALGORITHM

The navigation algorithm consists of eleven modular segments. The modules are divided and designated according to the function they perform. Modules I and II are initialization procedures and are one pass, while modules III through XI are organized in a linear loop with one subroutine called in module III. An explanation of each module can be found following.

Module I is the system initialization. The coordinates of the object transponders and any predesignated parameters are loaded into memory. All flags used through the algorithm are cleared.

Module II is the navigation computer initialization and window formation. The function of this module is to choose which hydrophone to ping, and to develop a history with which to initialize windows.

When the vehicle has been placed in the water, before it starts moving, each of the three hydrophones are consecutively pinged. The hydrophone with which the most returns are received is designated as the pinger. In case of a tie, the hydrophones have an ordered priority.

Once the pinger has been chosen, initialization of the windows can begin. The pinger is pinged several times in succession. When enough uniform data appears at a high enough frequency of occurance (for each element in the raw data matrix) then a window can be initialized using the most recent consistent sample. When windows have been formed for all available data returning, then initialization is complete. Control is passed into the loop starting with module III. The main system is signaled through the reliabililty word that initialization is over and the vehicle can begin motion.

Module III, the active navigation and data handler. First the "get data" subroutine is called. The "get data" subroutine controls the actual hardware to reset the counters, ping the chosen pinger, start the counters, test for counter overflow (no return), and load raw data (counter values) into the raw data matrix.

After the "get data" subroutine returns with the raw data matrix, the raw data is compared with the window and window size matrices and the coded matix is loaded accordingly. A "G" for a good return is coded as a five, a "B" for a bad return is coded as a three, and an "N" for no return is coded as a one, thus yielding a unique sum for any combination of returns in a row or column. When the coded matrix is completed control is passed to module IV.

In module IV, the computation decision module, the existing conditions are examined, and decisions are made to determine what information can be calculated. First the range is found to determine which information should be calculated. The coded matrix is examined and the most desireable pinger is chosen based on number of returns received in a row and previously set priorities (pinger one before pinger two before pinger three). If the previous pinger sum is good (two "G's" or more), then the criteria for position computations are checked. Two "G's" in the pinger row and history (last flags) or three "G's" in the pinger row (three flags) are the only situations where position can be computed. If position is computable, then the position flag is set and the last position flag (history availability from module XI) is checked. If the history exists, then velocity can be computed and the velocity flag is set.

Since the heading computation and error detection both require column sums of the coded matrix, these operations are performed together. If a column sum is checked and three "G's" are found, then the column is placed on the heading computation list. If three "G's" are not found and the vehicle is within the range between two hundred and one hundred feet then error detection is valid and can be performed. If three "B's" or three "N's" appear then hardware failure of the object transponder is indicated and recorded. If the column sum is a combination of returns, then the column is examined element by element. If the element has been coded anything oher than a "G", the hydrophone error is indicated and recorded.

After error detection is completed, the heading information is examined. The column which was used to compute the heading on the previous pass (stored in heading box) is checked. In order to maintain consistency for the relative heading, the heading column is not changed unless available information is sufficient. If the heading column is to be changed, the heading computation list is checked, and the column with the largest range (to provide for accuracy in the calculations) is chosen and its member is stored in the heading box. If three "G's" appear in any one column relative heading is computable and the relative heading flag is set. In addition, if the position flag is also set then the absolute heading can be computed, and the absolute heading flag is set.

Finally, the relative and absolute heading flags are checked along with their respective history flags to determine if the derivative calculations can be performed. If the information is available, then the "dot" flags are set.

Module V, position calculations, checks the position flag to determine if position is to be calculated, if so, the three flag is checked to determine which method of disambiguation is to be used. Once these flags are checked, appropriate action is taken and control is passed to module VI.

Module VI, the heading computation module, checks the relative heading and absolute heading flags. If these flags are set, then the heading box is checked to determine which column is to be used in the computations.

Module VII, the range and differential computations, checks the heading box, and uses the returns in that column to determine range to be consistent with the relative heading. Then the velocity and "dot" flags are checked (module IV) and the indicated calculations are performed.

Module VIII, result processing and window reset, checks and resets the windows and tabulates error counts. If there are any "G's" existing in a column, then the windows for non-good retuns in that column are based on the good returns with expanded sizes. The windows of any good return are immediately updated with the actual received return. Therefore, if a window is received for which no window exists, then it has been coded a "G" in module III and the window is automatically initialized in this module. If there are no "G's" existing in a column, then the windows remain the same but the window sizes are increased. If the window sizes are expanded beyond a given limit then the windows are cleared and will be initialized "on the fly" as previously explained.

If the position flag has been set, then there are at least two "G's" in a row in which to base the windows for the column. If the three flag is not set, then the third window can be computed upon which windows in that column can be based.

After the windows have been reset, the error analysis begins. The error flags from module IV are tabulated into two running counts. One count is for the object transponders, the other for the vehicle hydrophones. The running counts detect the number of errors over twenty four samples. If the number of errors exceed the prescribed limit the hardware malfunction flag is set.

Module IX, the reliability word formation checks flags set in module IX to determine which information will be sent to the main system computer. Bits are set in the reliability word which indicate the availability of this information. If the vehicle is within one hundred feet and the position flag is not set, then the history is checked. If history exists a bit in the reliability word is set to indicate old information. If history also is not available, then the old information bit is set, and bits are set for any history that does exist. The same procedure is followed for the vehicle being outside of the one hundred foot range with respect to the relative heading.

Module X, information loading and transmisison loads the appropriate information as indicated by the previously set reliability word into the proper memory addresses and sends this information to the main computer.

Module XI, flag processing and history manager checks flags set in module IV and loads the present information into history while setting the corresponding history availability ("last") flags. After history has been updated, all flags are cleared and control loops back to module III.

6.6 KNOWN DEPTH POSITION CALCULATION DERIVATION

Given: Coordinates of the three object transponders

A_x, A_y, A_z

B_x, B_y, B_z

C_x, C_y, C_z

Z coordinate of vehicle - V_z

Distance vehicle is located from each transponder

D_A, D_B, D_C

Find: x and y coordinates of vehicle (V_x, V_y)

Solution:

1) Project distances onto x-y plane to yield horizontal ranges, R_A, R_B, R_C

$$R_A = ((D_A)^2-(V_Z-A_Z)^2)^{1/2} \tag{1}$$
$$R_A = ((D_B)^2-(V_Z-B_Z)^2)^{1/2} \tag{2}$$
$$R_C = ((D_C)^2-(V_Z-C_Z)^2)^{1/2} \tag{3}$$

2) Compute horizontal distances between the object transponders.

$$D_{AB} = ((A_x-B_x)^2+(A_y-B_y)^2)^{1/2} \tag{4}$$
$$D_{AC} = ((A_x-C_x)^2+(A_y-C_y)^2)^{1/2} \tag{5}$$
$$D_{BC} = ((B_x-C_x)^2+(B_y-C_y)^2)^{1/2} \tag{6}$$

Using the law of cosines, d_1, the distance along the A-B baseline from transponder A to the intersection (x,y) with the normal from vehicle position to the A-B baseline is:

$$d_1 = \frac{(R_A)^2 + (D_{AB})^2 - (R_B)^2}{2D_{AB}} \qquad 7$$

Then by pythagorian's theorem, d_2, the normal distance from the vehicle to the baseline,

$$d_2 = ((R_A)^2 - (d_1)^2)^{1/2} \qquad 8$$

To find coordinates x,y, using similar triangles:

$$\frac{X-B_x}{d_1} = \frac{C_x-B_x}{D_{AB}} \quad , \qquad 9$$

$$X = B_x + \frac{d_1}{D_{AB}} (C_x-B_x) \quad . \qquad 10$$

and $$\frac{Y-B_y}{d_1} = \frac{C_y-B_y}{D_{AB}} \qquad 11$$

$$Y = B_y + \frac{d_1}{D} \quad (C_y-B_y) \quad . \qquad 12$$

since d_z is normal to the baseline, then its slope is given by,

$$\frac{V_y - Y}{V_x - X} = - \frac{C_x - B_x}{C_y - B_y} \qquad 13$$

and,

$$(V_x-X)^2 + (V_y-Y)^2 = d_2{}^2 \qquad 14$$

from equation 13,

$$V_y-Y = (V_x-X) \left[- \frac{C_x-B_x}{C_y-B_y} \right] \qquad 15$$

substituting equation 15 into equation 14:

$$(V_x-X)^2 + \left[(V_x-X)\left[-\frac{C_x-B_x}{C_y-B_y}\right]\right]^2 = d_2^2 \qquad 16$$

factoring $(V_x-X)^2$ out of the large brackets,

$$(V_x-X)^2 \left[1 + \left(\frac{C_x-B_x}{C_y-B_y}\right)^2\right] = d_2^2 \qquad 17$$

solving equation 17 for V_x:

$$V_x = \left[\frac{d_2^2}{\left[1 + \left(\frac{C_x-B_x}{C_y-B_y}\right)^2\right]}\right]^{1/2} -x \qquad 18$$

But,

$$\frac{1}{1 + \left(\frac{C_x-B_x}{C_y-B_y}\right)^2} = \frac{(C_y-B_y)^2}{(C_y-B_y)^2 + (C_x-B_x)^2} \qquad 19$$

then from equation 6,

$$\frac{1}{1 + \left(\frac{C_x-B_x}{C_y-B_y}\right)^2} = \frac{(C_y-B_y)^2}{D_{AB}^2} \qquad 20$$

substituting equation 20 into equation 18,

$$V_x = \left[\frac{d_2^2 (C_y-B_y)^2}{D_{AB}^2}\right]^{1/2} -x \qquad 21$$

and,

$$V_x = x \pm \left(\frac{C_y-B_y}{D_{AB}}\right) d_2 \qquad 22$$

substituting equation 10 into equation 22,

$$V_x = B_x + \left(\frac{C_x - B_x}{D_{AB}}\right) d_1 \pm \left(\frac{C_y - B_y}{D_{AB}}\right) d_2 \qquad 23$$

Similarly, the corresponding solutions for V_y are:

$$V_y = B_y + \left(\frac{C_x - B_x}{D_{AB}}\right) d_1 \mp \left(\frac{C_y - B_y}{D_{AB}}\right) d_2 \qquad 24$$

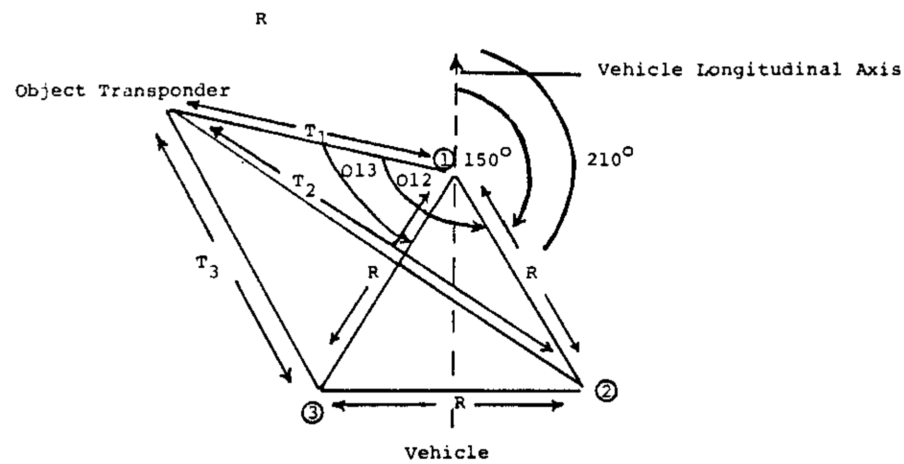
Note that there are two sets of ambiguous solutions, therefore necessitating disambiguation by reference to a third transponder or history.

6.7 RELATIVE AND ABSOLUTE HEADING COMPUTATION DERIVATION

Given: Distances from each of the vehicle hydrophones to one of the three object transponders:

$$T_1, T_2, T_3$$

Distance between hydrophones (length of side of equilateral triangle):



From the law of cosines:

$$O_{12} = \cos^{-1}\left[\frac{R^2+T_1-T_2{}^2}{2RT_1}\right] \qquad 25$$

and,

$$O_{13} = \cos^{-1}\left[\frac{R^2+T_1{}^2-T_3{}^2}{2RT_1}\right] \qquad 26$$

for O_{12}, the relative heading, O_{REL}, is,

$$O_{REL} = 210^{\circ} \pm O_{12} \qquad 27$$

and for O_{13},

$$O_{REL} = 150^{\circ} \pm O_{13} \qquad 28$$

From equations 27 and 28 there are four possible solutions, however, one solution from 27 must correspond to one solution from 28 since both equations are attempting to describe the same angle (O_{REL}). By determining which solutions are closest, a disambiguous solution can be chosen.

Four differences:

$$(150^{o}+O_{12})-(210^{o}+O_{13})=(60^{o}+O_{12}-O_{13}) \qquad 29$$

$$(150^{o}+O_{12})-(210^{o}-O_{13})=(60^{o}+O_{12}+O_{13}) \qquad 30$$

$$(150^{o}-O_{12})-(210^{o}+O_{13})=(60^{o}-O_{12}-O_{13}) \qquad 31$$

$$(150^{o}-O_{12})-(210^{o}-O_{13})=(60^{o}-O_{12}+O_{13}) \qquad 32$$

By examining the magnitudes of equations 29 through 32, the minimum difference can be found. By averaging the two solutions from equations 27 and 28, which constitute the minimum difference, the nonambiguous relative heading is found.

To find the absolute heading the vehicle coordinates V_x, and V_y must be known as well as the transponder coordinates T_x, T_y.

To correct the relative heading from the angle which is between the vehicle longitudinal axis and the connecting line between the object transponder and the vehicle, to the angle between the vehicle longitudinal axis and the x-axis:

$$\alpha = \mathrm{TAN}^{-1}\left[\frac{V_y-T_y}{T_x-V_x}\right] \qquad 33$$

Equation 33 is self adjusting. The sign of the resulting angle will add or subtract to the relative heading to justify it to the 'closest' x-axis.

Then depending upon whether the transponder or the vehicle is closest to the y-axis,

if $(T_X - V_X)$ is posistive,

$$O_{ABS} = O_{REL} + \alpha \qquad 34$$

if $(T_X - V_X)$ is negative,

$$O_{ABS} = 180^{\circ} - (O_{REL} + \alpha) \qquad 35$$

6.8 CONSIDERATIONS ON ACOUSTIC NAVIGATION IN A REVERBERANT ENVIRONMENT

During the previous year, a navigation system concept has been defined which uses acoustic ranging to obtain position data. This data depends on the accurate determination of the arrival times of acoustic pulses. Unfortunately multipath and reverberation distort the time of arrival of acoustic pulses. The proposed testing program in 1980-81 is meant to bound the multipath problem within the test area as it relates to the navigation system.

For the case of digital pulse transmission in the ocean, multipath causes the received signal at a point to be comprised of a main pulse plus reflections of the main pulse, which are modified in amplitude and phase. We have chosen to represent this signal in two ways.

In the time domain:

$$v(t) = A_o \, \pi \left(\frac{T}{t-D_o} \right) \cos W_o t + \sum_{i=1}^{n} A_i \, \pi \left(\frac{T}{t-D_i} \right) \cos \left(W_o (t-D_i) \right) \qquad 36$$

A is the relation amplitude of each pulse and $\pi\left(\frac{T}{t}\right)$ is a rectangular pulse of duration T which starts when $t-D_i$ is zero. Di is the delay associated with each reflection.

The first term in this equation is the signal obtained from the first pulse. The terms under the summation sign include all of the signals for n reflections.

In the frequency domain:

$$V(F) = A_o \text{ sinc } (W-W_o)T + \sum_{i=1}^{n} \{A_i \text{ sinc } (W-W_o)\pi\} e^{jwDi} \qquad 37$$

where the sinc function is the familiar (sin(x))/x function. One way of interpretting this equation is to consider the total signal spectrum as the spectrum due to the direct pulse plus modifications to this spectrum due to all of the reflections.

The EAVE navigation system requires the detection of acoustic pulses which contain two pieces of information. First is the distance from a point to a transponder. Second is the identity of that transponder. To determine which transponder a pulse had as its source, the pulses must be orthogonal. We will consider two methods of accomplishing this; distinct in time; distinct in frequency. This leads us to three alternatives for receiver design.

Time Diverse System

When all transponders reply on a single frequency, the transponders must reply in a way which will allow the source transponder to be uniquely identifiable. For all transponders to be uniquely identifiable, each transponder must answer in a specific window in time as determined by the maximum travel time of the acoustic pulse. The windows are defined with respect to the first transponder, which may answer at any time. For N transponders, there are N-1 windows and the width of the window is maximum distance divided by speed.

Thus, the time to complete a full cycle is N times the duration of the window. However, because of multipath, the second term in equation 36 is non-zero. This indicates that energy is spread in time from one window to another. To avoid this, guard bands can be defined which allow the attenuation of acoustic energy to a level below the system noise threshold. This means that reply time of the transponders is staggered at predetermined intervals.

Fortunately, the system receiver gain may be set on the first pulse that arrives. Any reflections of the first pulse will have traveled over a longer path and suffered greater attenuation. The time required for guard bands may be calculated from the transmission loss equations. Because the start of the pulse window is known, the AGC may be reset at the beginning of each window, so that the signal level will not depend on the previously received pulse.

Frequency Diverse Systems

Frequency diverse systems have the advantage of interrogating all transponders at once. The time to complete a full cycle is the time it takes to complete one round trip. This system may announce a new position in approximately 1/n times the single frequency system where n is the number of transponders.

The transponders are identified by band pass filtering the received signal spectrum around the transponder reply frequencies.

From equation 37, it is seen that the spectrum for the acoustic pulse with multipath is not well known. The second term here causes energy to be spread in an unknown fashion. The effects of multipath make it difficult to know what guard bands in frequency are required.

Frequency and Time Diverse Systems

It is possible to make a system in which the return pulses are orthogonal in both time and frequency. This system further insures that the proper pulses are associated with the proper transducer. The cost of this insurance is that the system is more complex. This system requires that both the time domain and frequency domain components of the signal be well known.

Testing Program

Before proceeding further with the navigation system design, some properties of multipath at the frequencies of interest must be measured. This data is required to help quantify the time dispersion and frequency dispersion of the acoustic pulses. With this knowledge, appropriate design decisions can be made.

Some of the design decisions include:

1) Given a frequency diverse system, how great is the spectral distortion, due to multipath, within the operating area?

2) What spectral guard bands are required for a frequency diverse system?

3) Is the channel better suited to time orthogonal systems at these frequencies?

4) What guard bands in time are required?

These questions all impact the systems performance characteristics.

The proposed test consists of recording several acoustic pulse records for different system geometries then processing these returns to evaluate the signal spectrum of the return pulse. The time required for guard bands for a single frequency system will also be taken from these records. The data from this test is imperative for the next phase of system design.

7.0 AN EXAMINATION OF VEHICLE DYNAMICS

The Experimental Autonomous Vehicle (EAVE), presently under development at the University of New Hampshire, has taken on the task of sending the EAVE vehicle inside a complex structure to perform inspection tasks. This requires that the vehicle have an accurate control system which will allow the vehicle to move about inside a structure, without colliding with the structure, becoming entangled in cables, etc.

In order to develop a control system, it is necessary to have a mathematical model which describes how the vehicle moves. This section describes the development of such a model. The model assumed is a constant coefficient nonlinear model. The constant coefficient approach is taken to simplify the development of the model. Later, the constant coefficients will be replaced by functions which more accurately describe the vehicle's dynamics.

The vehicle is assumed symetric in that it response is identical whether it is traveling in a positive or negative direction. Further, it is assumed that there are no hydrodynamic lift forces generated by the vehicle. This is a reasonable assumption in that all of the vehicle components are symetric in shape, there are no wing-like parts.

7.1 THE MODEL COORDINATE SYSTEMS

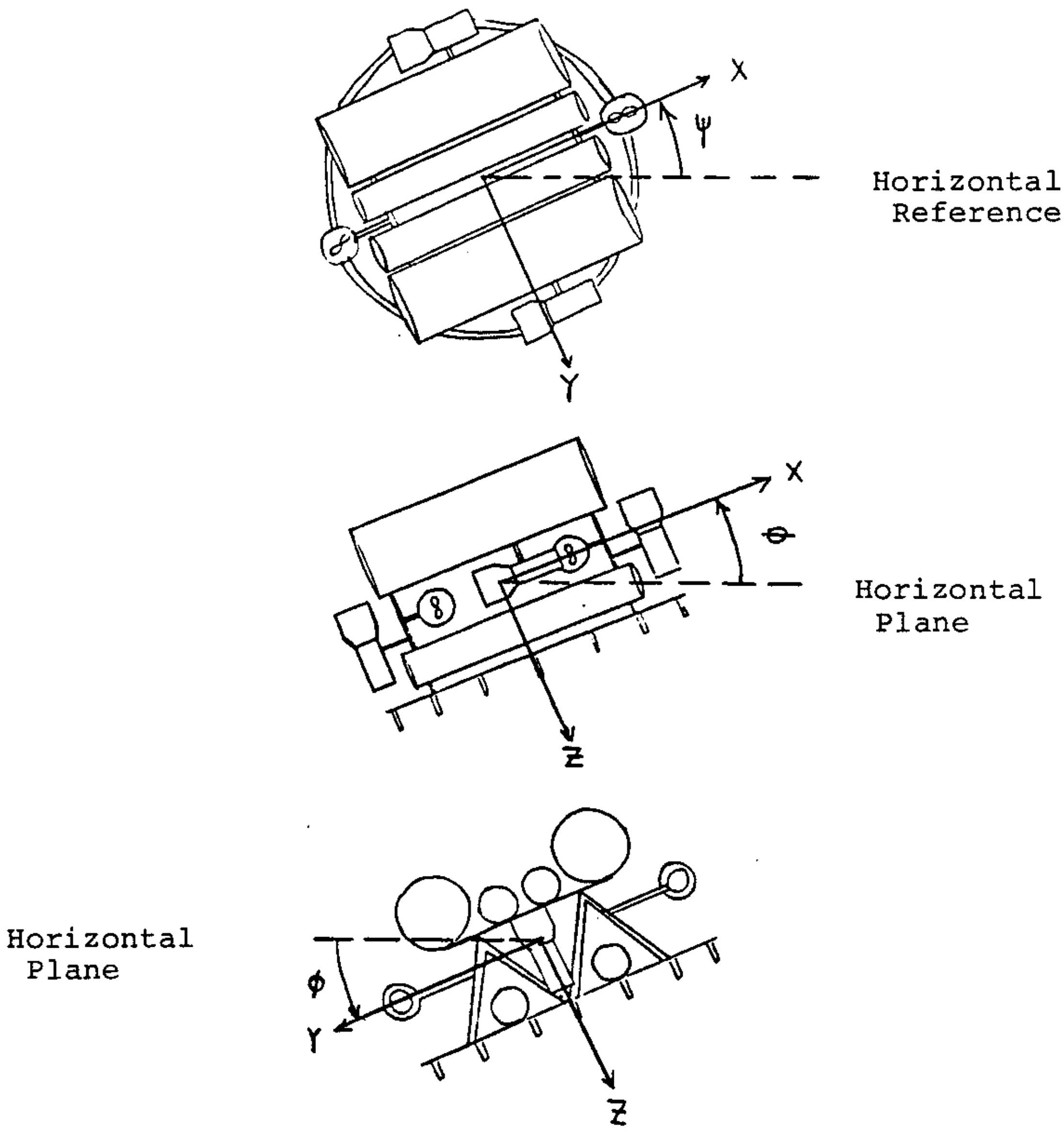
Two coordinate systems are required to model the vehicle's motion. The first is a coordinate frame which is located on the vehicle, with its origin at the center of gravity (CG). The second is a global reference frame, in which the vehicle coordinate frame moves. Both coordinate systems are Right Hand Cartesian systems.

The vehicle coordinate system has its x axis point out of the bow, the y axis point out of the starboard side, and the z axis points down through the keel. The global coordinate system has two axes in the horizontal plane and one axis which points down. One of the horizontal plane axes (X_g) may be oriented in the North direction. The downward pointing axis is labeled Z_g.

With the origins of each coordinate system co-located, angular definitions between the systems are made as shown in Figure 14. Accelerations on the origin of the vehicle's coordinate system in the global reference frame are the accelerations seen along the vehicle system axis, rotated through the roll, pitch and yaw angles as defined in Figure 14. The transformation matrix which performs this rotation is given in Equation 38.

Figure 14

BODY AXIS AND ANGULAR DEFINITIONS



$$T = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & \sin\theta \\ \sin\theta\sin\phi\cos\psi - \cos\phi\sin\psi & \sin\theta\sin\phi\sin\psi + \cos\phi\cos\psi & -\cos\theta\sin\phi \\ -\sin\theta\cos\phi\cos\psi - \sin\phi\sin\psi & \sin\phi\cos\psi - \sin\theta\cos\phi\sin\psi & \cos\theta\cos\phi \end{bmatrix} \qquad 38$$

Thus,

$$\begin{bmatrix} \ddot{X}_0 \\ \ddot{Y}_0 \\ \ddot{Z}_0 \end{bmatrix} = T \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} \qquad 39$$

In a similar manner, accelerations in the global frame are transformed into the vehicle frame by a similar transformation matrix, given in Equation 40.

$$D = \begin{bmatrix} \cos\theta\cos\psi & -\cos\theta\sin\psi & -\sin\theta \\ \cos\phi\sin\psi + \sin\theta\sin\phi\cos\psi & \cos\phi\cos\psi - \sin\theta\sin\phi\sin\psi & \cos\theta\sin\phi \\ \sin\theta\cos\phi\cos\psi - \sin\phi\sin\psi & -\sin\theta\cos\phi\sin\psi - \sin\phi\cos\psi & \cos\theta\cos\phi \end{bmatrix} \qquad 40$$

Thus,

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} \ddot{X}^1 \\ \ddot{Y}^1 \\ \ddot{Z}^1 \end{bmatrix} + D \begin{bmatrix} 0 \\ 0 \\ W-B \end{bmatrix} , \qquad 41$$

which states that the acceleration on the vehicle's coordinate frame is that of the system inputs and states (prime notation) plus the global acceleration due to the difference between the gravitational and buoyant force transformed through the D matrix.

Integrating the accelerations on the origin of the vehicle's coordinate system as seen in the global reference frame twice gives the position of the origin of the vehicle system in the global system.

That is:

$$\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} = \frac{1}{s^2} I \begin{bmatrix} \ddot{X}_0 \\ \ddot{Y}_0 \\ \ddot{Z}_0 \end{bmatrix} \qquad 42$$

To determine the vehicle's position in the global reference frame consists of the following steps. First calculate the accelerations on the vehicle as a function of the system states and gravity rotated through the D matrix. Second, rotate this set of accelerations through the T matrix to get the global reference frame accelerations. Third, integrate these accelerations twice to get the vehicle's position.

7.2 DEVELOPMENT OF A GENERAL FORM OF VEHICLE MODEL

The forces which act on the vehicle in the vehicle's reference frame are developed here. The model is not a general submersible model but rather a model which has inputs and reactive forces which are specific to a vehicle shaped like the EAVE vehicle. That is, thrusters are oriented as they are on the EAVE system, the frame is considered tubular in form, etc. The response of the vehicle is broken down into six motions. They are 3 translations along each of the vehicle's 3 axes and 3 angles, roll, pitch and yaw, developed between the vehicle's coordinate system and the reference.

Response along the X Axis

The input forces to the vehicle system along the x axis are due to thrusters 1, 2, 3, and 4, with the thrusters defined as in Figure 14. It is assumed that thrusters 5 and 6 do not couple into motion along the vehicle's x axis. Thrusters 3 and 4 couple directly into motion along x. A small part of thrusters 1 and 2 couples into motion along x, the scaling factor for this couple is designated KS. The reactive forces are due to the resistance of the vehicle's mass to acceleration (inertia) and hydrodynamic drag. The differential equation which describes this is:

$$U3 + U4 + KS(U2-U1) - M_x\ddot{X} - D_x\dot{X} = 0 \qquad 43$$

where M_x is the mass coefficient along the x direction and D_x is the drag coefficient along the x direction. Using the S operator to denote differentiation, the following is obtained:

$$(M_xS^2 + D_xS)X = KS(U2 - U1) + U3 + U4 \qquad 44$$

This system block diagram is shown in Figure 15.

Response along the y axis

The input forces along the y axis are due to thrusters 5 and 6. It is assumed that no other thrusters couple into motion along the y axis. The reactive forces are due to inertia and drag. The differential equation that describes this motion is

$$U5 + U6 - M_y\ddot{Y} - D_y\dot{Y} = 0 \qquad 45$$

or using s operator notation,

$$U5 + U6 = (M_yS^2 + D_yS)\ Y \qquad 46$$

In equations 45 and 46, M_y is the inertial mass coefficient and D_y is the drag coefficient for motion along the y axis. The block diagram for this system is shown in Figure 16.

Figure 15

BLOCK DIAGRAM FOR MOTION ALONG X AXIS

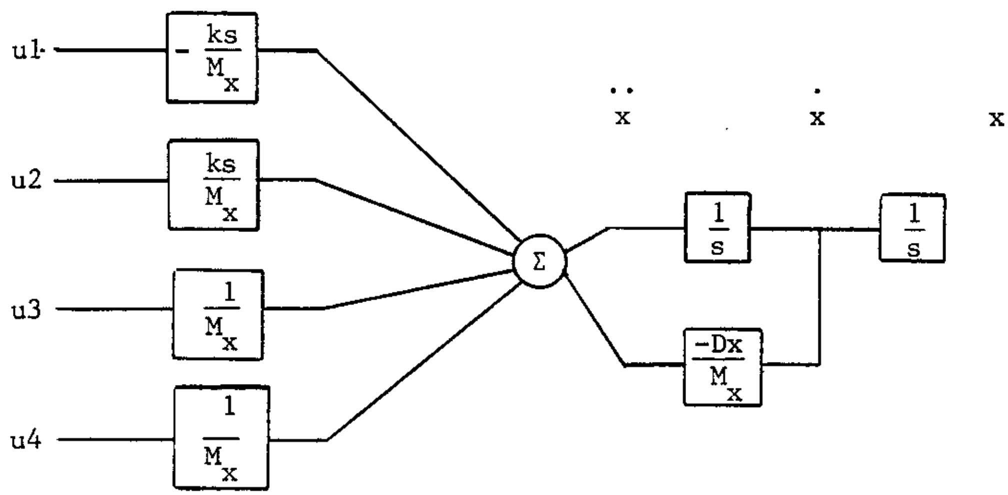
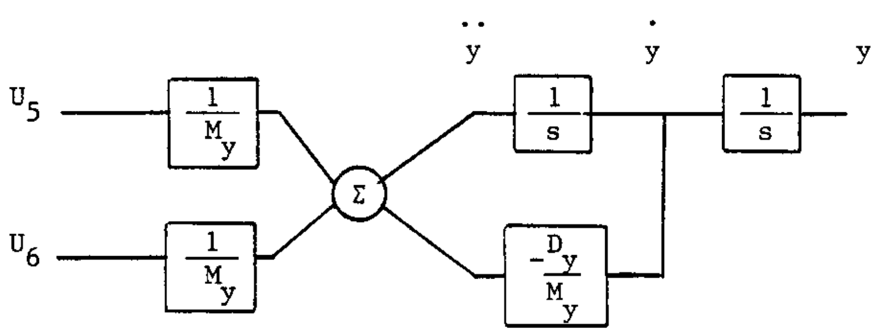


Figure 16

BLOCK DIAGRAM FOR MOTION ALONG Y AXIS



Response along the Z axis

The input forces along the z axis are due to U1 and U2. It is assumed that no other thruster couples into motion along the z axis. The reactive forces are again due to inertia and drag. The z axis differential equation is

$$U1 + U2 - M_Z \ddot{Z} - D_Z \dot{Z} = 0 \qquad 47$$

Similarly using operator notation,

$$U1 + U2 = (M_Z S^2 + D_Z S)\ Z \qquad 48$$

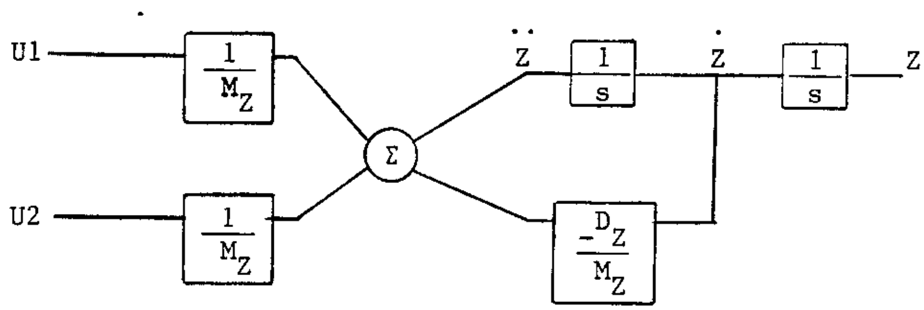
The block diagram for this system is shown in Figure 17.

Angular Response of EAVE

Three angles are used to describe the orientation of the vehicle reference frame. The angles are called Pitch, Roll, and Yaw. The Pitch angle is the angle between the x axis of the vehicle system and the horizontal plane. The roll angle is the angle between the y axis of the vehicle and the horizontal plane. Both the pitch and roll angles are stabilized by the separation of the center of buoyancy (CB) and the center of gravity (CG). The yaw angle is the angle measured between the global X_0 axis and the vehicle x axis in the horizontal plane. All rotations take place about the CG.

Figure 17

BLOCK DIAGRAM FOR MOTION ALONG Z AXIS



The principle forces that cause angle generations are the thruster forces and the drag forces of translation, each of these is multiplied by its respective moment arm to produce a torque about the CG. The reactive forces are due to rotational inertia, rotational drag, and where appropriate, the separation of the CG and CB.

Rotation about the x Axis - Roll

The positive definitions of force directions and moment arms are shown in Figure 18. The causal forces for rotation are the drag force along the z and y axis and the thrusters #5 and #6. The reactive forces are rotational inertia, rotational drag and the CB-CG moment. The differential equation which describes this is

$$Jx\ \ddot{\phi} + Bx\ \dot{\phi} + L\ G \sin\phi + Hl\ (U5 + U6) + Dzy * Dz * \dot{Z} - Dyz * Dy * \dot{Y} = 0 \qquad 49$$

Using 5 operator notation, equation 49 becomes

$$(Jx\ s^2 + Bxs)\ \phi + L\ G \sin\phi = Dyz * Dy * sy - Hl(U5 + U6) - Dzy * Dz * sz \qquad 50$$

The system block diagram which describes the roll angle is shown in Figure 19.

Figure 18

X AXIS ROTATIONAL FORCE DIAGRAM

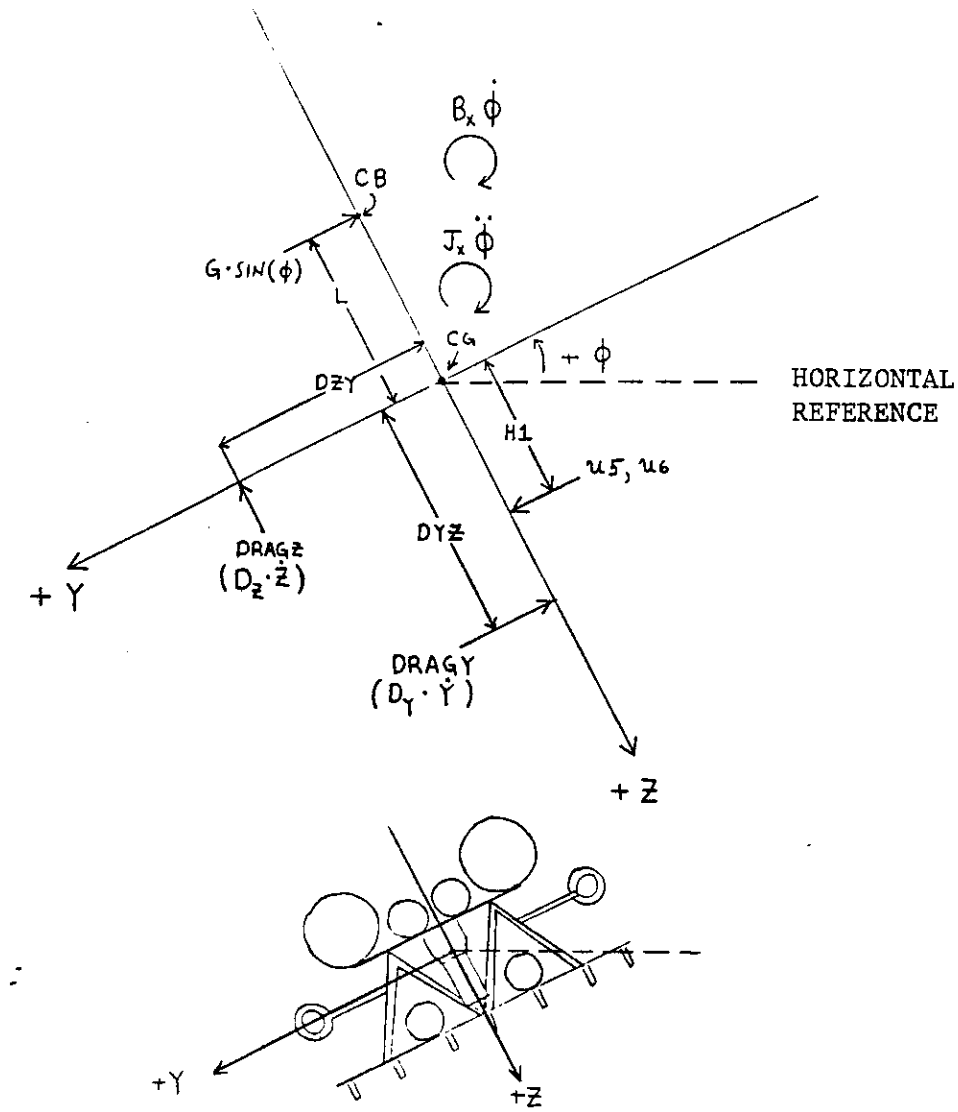
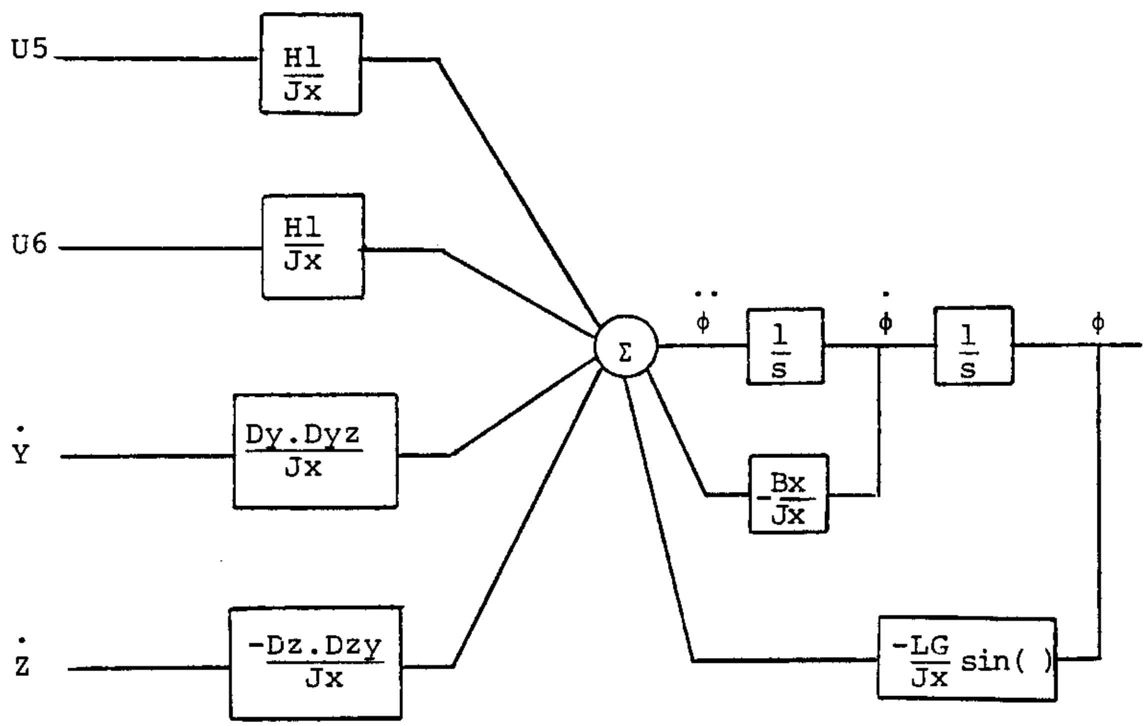


Figure 19

ROLL ANGLE SYSTEM BLOCK DIAGRAM



Rotation about the y Axis - Pitch

The forces which are assumed to cause the vehicle to pitch are due to translational drag along the x and z axis as well asa thrusters 1, 2, 3, and 4, each multiplied by its respective moment arm. The reactional forces to this rotation are the rotational drag and rotational inertia terms, as well as the force due to the separation of the CB and CG. The definitions of positive forces and moments are shown in Figure 20.

The differential equation which describes this motion is

$$Jy\ddot{\theta} + By\dot{\theta} + LG\ \sin\theta + U1*A1 + Dxz * Dx * \dot{X} - U2 * A^2 - H2(U3 + U4) - Dzx * Dz * \dot{Z} = 0 \qquad 51$$

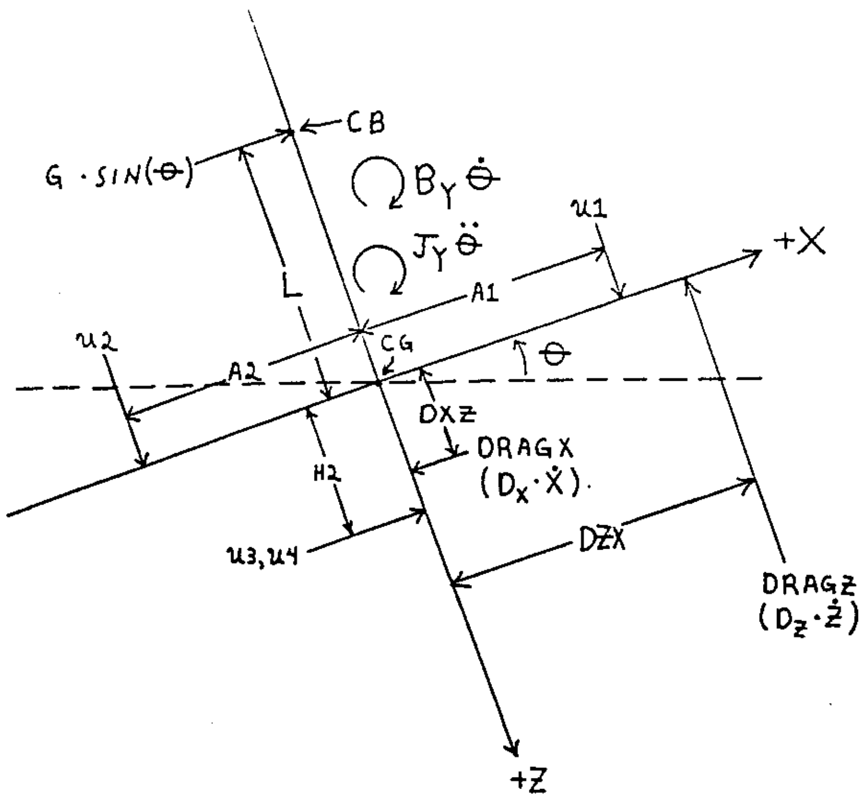
Using s operator notation, equation 14 becomes

$$(Jy\ s^2 + By\ s)\ \theta + LG\ \sin\theta = A2 * U2 - A1 * U1 + H2(U3 + U4) + Dzx * Dz * Sz - Dxz * Dx * Sx \qquad 52$$

The system block diagram is shown in Figure 21.

Figure 20

Y AXIS ROTATIONAL FORCE DIAGRAM



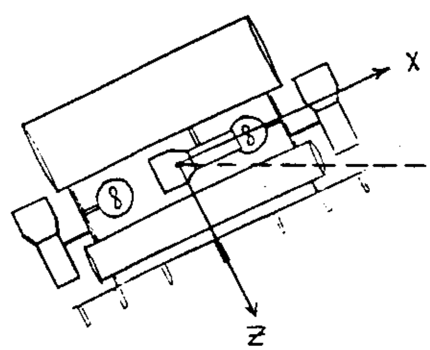
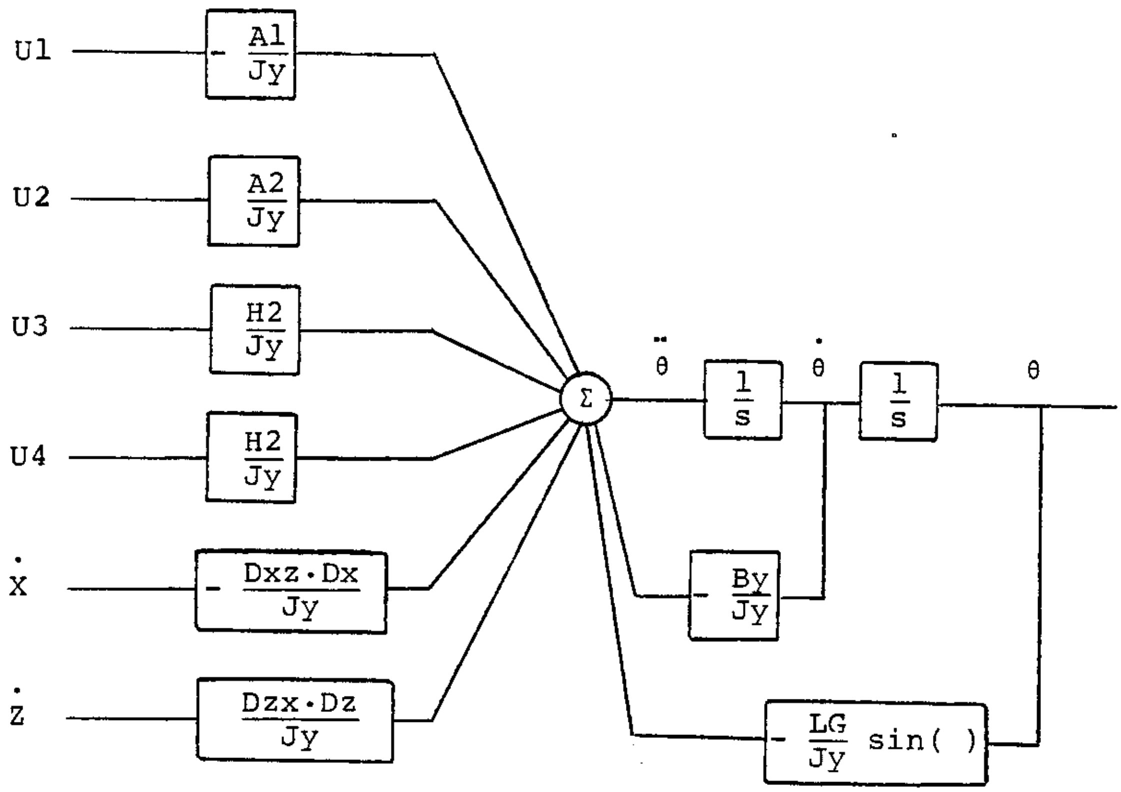


Figure 21

PITCH ANGLE BLOCK DIAGRAM



Rotation about the z Axis - Yaw

The forces that cause the vehicle to rotate about the z axis are the translational drag forces along the x and y axis and all of the thrusters. Each force has its respective moment arm. The reactive forces are due to rotational inertia and rotational drag. Positive force and moment arm definitions are shown in Figure 22.

The differential equation which describes this motion is

$$Jz\ddot{\Psi} + Bz\ddot{\Psi} + KS(A1 \cdot U1 + A2 \cdot U^2) - A3 \cdot U3 + A4 \cdot U4 + A5 \cdot U5 - A6 \cdot U6 + Dxy \cdot D \cdot \dot{X} - Dyx \cdot Dy \cdot \dot{Y} = 0 \qquad 53$$

Using s operator notation, Equation 53 becomes

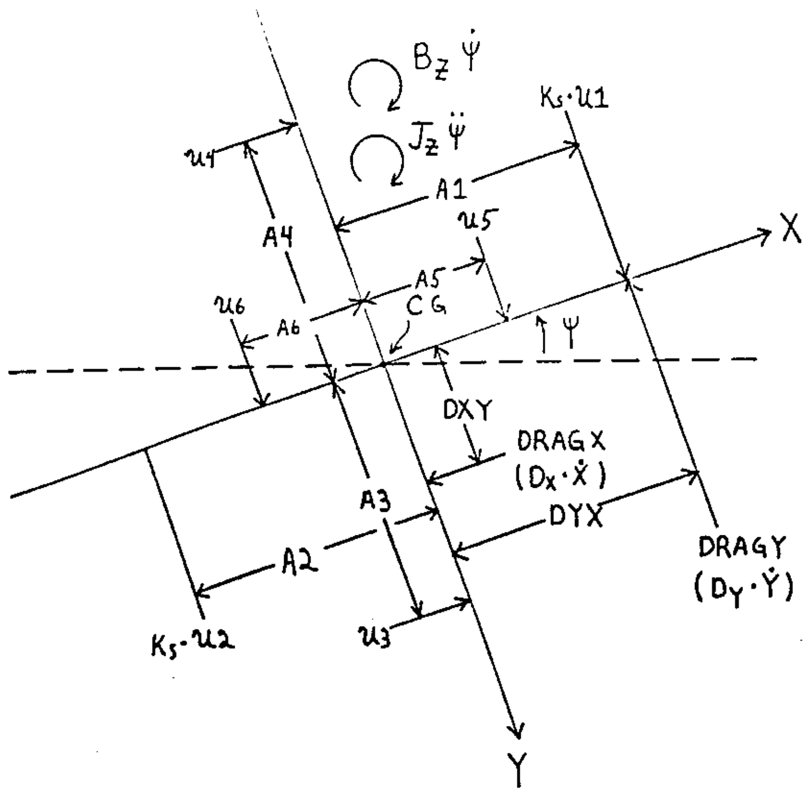
$$(Jz\ S^2 + Bz\ S)\,\Psi = -KS(A1*U1 + A2*U2) + A3*U3 - A4*U4 - A5*U5 + A6*U6 - Dxy*Dx*Sx + Dyx*Dy*Sy \qquad 54$$

The system block diagram for this is shown in Figure 23.

This concludes the development of the equations of translation along each axis in the vehicle coordinate frame and the equations which describe the angular response between the vehicle's coordinate system and the reference frame. What remains is to take all of the equations and tie them together into a comprehensive model.

Figure 22

Z AXIS ROTATIONAL FORCE DIAGRAM



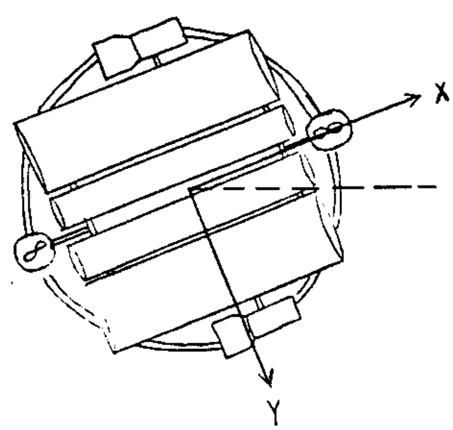
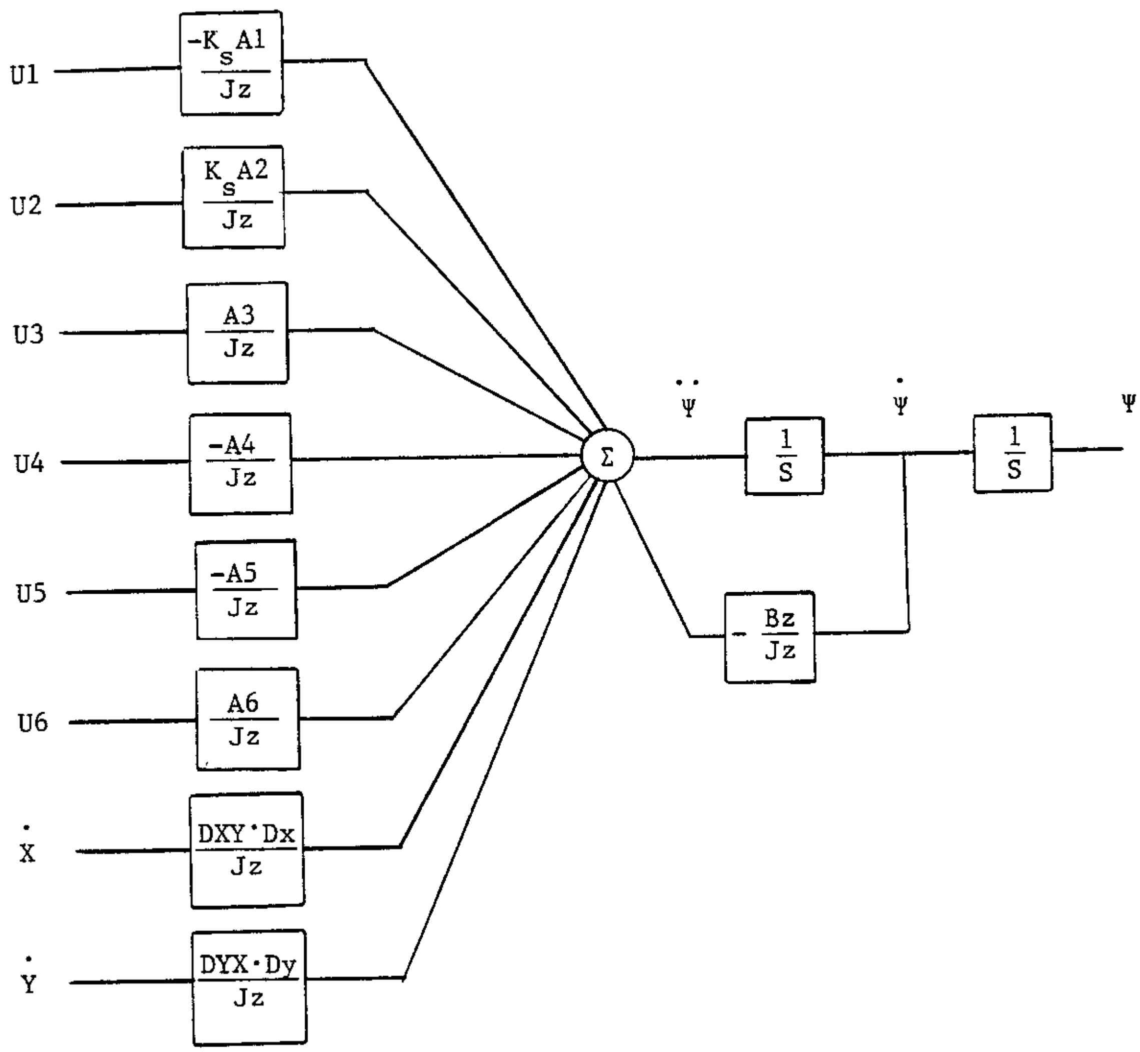


Figure 23

YAW ANGLE SYSTEM BLOCK DIAGRAM



7.3 STATE EQUATIONS FOR THE EAVE DYNAMIC MODEL

The state equation representation of the EAVE dynamic model is made by making the following state definitions, equating physical parameters to state variables.

State Variable	Physical Parameter
X1	X - Position in X
X2	$\dot{x}$ - Velocity in X
X3	Y - Position in Y
X4	$\dot{y}$ - Velocity in Y
X5	Z - Position in Z
X6	$\dot{z}$ - Velocity in Z
X7	ϕ - Roll about X
X8	$\dot{\phi}$ - Velocity of roll about X
X9	θ - Pitch about Y
X10	$\dot{\theta}$ - Velocity of pitch about Y
X11	Ψ - Yaw about Z
X12	$\dot{\Psi}$ - Velocity of yaw about Z

From these definitions, the following matrix equation is obtained using equations 38 through 54. Here all of the states of the vehicle system are coupled together in one large matrix. The interdependence of the states upon the other system parameters (coupling) is readily apparent.

This concludes the development of the general form of the EAVE dydnamic model. What remains is to implement the model in some form of simulation package and verify the general form of the model. The verification will consist of applying various input forces to the system and checking the response to see if it is in the proper direction. This is presently underway.

Equation 55

$$
\begin{bmatrix} X1 \\ X2 \\ X3 \\ X4 \\ X5 \\ X6 \\ X7 \\ X8 \\ X9 \\ X10 \\ X11 \\ X12 \end{bmatrix}
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{Dx}{Mx} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{Dy}{My} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{Dz}{Mz} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{Dy \cdot Dyz}{Jx} & 0 & \frac{Dz \cdot Dzy}{Jx} & -\frac{LG}{Jy}\sin & -\frac{Bx}{Jx} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & \frac{Dx \cdot Dxz}{Jy} & 0 & 0 & 0 & \frac{Dz \cdot Dzx}{Jy} & 0 & 0 & -\frac{LG}{Jy}\sin & -\frac{By}{Jy} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & \frac{Dx \cdot Dxy}{Jz} & 0 & \frac{Dy \cdot Dyx}{Jz} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{Bz}{Jz}
\end{bmatrix}
\begin{bmatrix} X1 \\ X2 \\ X3 \\ X4 \\ X5 \\ X6 \\ X7 \\ X8 \\ X9 \\ X10 \\ X11 \\ X12 \end{bmatrix}
+
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{Ks}{Mx} & \frac{Ks}{Mx} & \frac{1}{Mx} & \frac{1}{Mx} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{My} & \frac{1}{My} \\
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{Mz} & \frac{1}{Mz} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{H1}{Jx} & \frac{H1}{Jx} \\
0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{A1}{Jy} & \frac{A2}{Jy} & \frac{H2}{Jy} & \frac{H2}{Jy} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{KsA1}{Jz} & -\frac{KsA2}{Jz} & \frac{A3}{Jz} & -\frac{A4}{Jz} & -\frac{A5}{Jz} & \frac{A6}{Jz}
\end{bmatrix}
\begin{bmatrix} U1 \\ U2 \\ U3 \\ U4 \\ U5 \\ U6 \end{bmatrix}
$$

8.0 COMMUNICATION SYSTEM

The communication system attempts to exchange command and control information with the vehicle at teletype data rates, and at ranges comparable to those likely to be encountered in the structural inspection case. The shallow water range at Lake Winnipesaukeke, as well as the confines of a structure represent two of the worst possible acoustic communication environments, due to their high multipath. Studies of the data link, which are incomplete at this writing, are employing frequency diversity, and error correcting codes, to attempt to cope with the multipath. Communication with the vehicle appears to be essential in many foreseen missions, and a reliable approach must be found to exchange data between vehicle and operator. It is observed, however, that the structural inspection task does not present an urgent need for communication of status information, and thus in the initial developmental phase, communication should rightly take less priority than the other system problems. It is observed further that the top priority navigation problem will encounter acoustic paths identical to that of the communication system which will prove instructive to subsequent communication system design.

During 1979-1980, system development on communication was set aside in order to place greater resources on acoustic navigation. Demonstration has been made of teletype (300 baud) communications over modest path lengths in relatively shallow water, and the work at the Naval Ocean Systems Command has certainly demonstrated that slow-scan television may be transmitted over substantial ranges in a near-ideal environment. The consequences of intra-structure reverberation, however, promises to make useful communication bandwidth quite difficult to obtain. The modelling of an adaptive communications channel to accommodate structural multipath will profit greatly from the experimental work planned in 1981 for navigation system development. Further work on communications with the vehicle, for the structural inspection mission has been put aside for a year in order to benefit from the navigation system acoustic studies.

9.0 REVIEW OF NEXT STEPS

The goal of a 1980-1981 program is simply expressed. We plan to establish a partial prototype three-dimensional structure of the floor of Lake Winnipesaukee, and to cause the EAVE vehicle to transit through that structure, to perform the simple inspection task of picture taking, and to return to the launching site. All of this is to be accomplished under on-board software control, using data acquired by the vehicle while in motion. Stated differently, we wish to demonstrate fully autonomous vehicle performance in three-dimensions, even as a single dimensioned autonomous task was performed in 1979. The goal of this program, as expressed by the U.S. Geological Survey, is the development of technology that may be applied to offshore petroleum related inspection problems rather than the development of an operational vehicle. The tasks offered here are consistent with that philosophy.

The accomplishment of these complex goals, with this modestly-sized effort, is not simple, and several key problems must be solved to achieve satisfactory results. Six interrelated technical studies are anticipated as follows:

1) Provide a control system design capable of enabling the vehicle to maneuver precisely within the confines of an offshore structure.

2A) Fabricate a complete navigation system, based on a design of three receivers in the vehicle, each tracking a remote transmitter.

2B) Prepare for multiple-system redundancy in navigation by supplying five transponder systems.

2C) Study alternative systems of navigation, including time in addition to frequency multiplexing, as well as a combined acoustic and inertial system.

3) Define the software necessary to serve the needs of each of the three sub-systems, Control, Navigation, and Communications, by integrating their needs with the vehicle's operating sytem.

3A) Develop a multi-tasking real-time operating system capable of coordinating and controlling the operation of three-subsystem CPU's and/or their operating sytems.

3B) Define and implement an efficient and deadlock proof scheme of interprocessor communication.

3C) Integrate the software system with the requirements of the mission.

4) Maintain the Operating System, and provide source listings for the Inner Processors Input/Output systems.

5) Install the communiction link that was previously developed. Modify the link as is necessary to learnings from the system tests.

6A) Install a test structure in Lake Winnipesaukee, constructed of six inch pipe.

6B) Construct an acoustic test facility to support aoustic subsystem components in mid-water to evaluate propagation problems within the structure.

6C) Conduct tests of the vehicle with the purpose of a demonstration of the autonomous transiting of the vehicle through the structure.

- Subsequent Program Development

The effort reported in this document is highly focused, aimed at completing the assignments indicated on the Milestone Chart. If all the technical development problems are surmounted, there will be a successful demonstration of the vehicle's ability to navigate through a simple structure, and to accomplish proof of its skill by the performance of a simple task.

The accomplishment of next year's program, however, leaves much to be done to bring the technology to applicable form for field usage. There remain serious gaps that must be resolved.

They include:

- Visual imaging systems require clear water and an ample source of illumination. Short range acoustical imaging systems may be useful in murky water. With short pulse transmission, gated imaging is possible, which sections the image in range. This approach may be worthy of further study.

- A visual sensing system is highly desirable. The application of an unmanned vehicle is limited unless it will serve as the eyes of a human being who can apply his judgment to the observations of the vehicle sensors. Photographic evidence, delivered after the vehicle returns, will be most useful for some environments. It is not sufficient for all, however, and a camera and a real time link should be available. Such developments must receive attention.

- The acoustic navigation system must be expanded to include a combination inertial, doppler, and ranging system if precise position keeping for close inspection work is contemplated. As an alternative, a structurally referenced navigation should be considered, where the vehicle computes its position in totally local terms of a work station situated a given distance from a known structure feature.

- Sensors should be examined that extend the type of knowledge available to the vehicle. These could include tactile sensors to locate physical boundaries, and a pointing sonar to derive range and bearing to potential interferences.

- The question of communications for data purposes has deliberately been made subordinate to the navigation system in this year's work. The role of data communications must be re-examined and implemented as required.

- Communications with the Petroleum Industry must be more vigorous. This possibly could mean a joint test of the vehicle, as it will evolve from the 1980-1981 program, at a convenient site in the Gulf of Mexico. The terms of the tests must be compatible with the developmental nature of the vehicle, and the status of its progress at that time.

There are undoubtedly many other potential steps that are consistent with the state of the art, and the developments that have been achieved by the Laboratory. In due course, these considerations will be placed on paper for further discussion.

Appendix A

STRUCTURE ANALYSIS CALCULATIONS

During the structure analysis several calculations are performed quite often, are essential to the system, and must be efficient. These calculations include:

Distance calculations:

point to point (in 2 and 3 space)

point to line (in 3 space)

line to line (in 3 space)

point to plane (in 3 space)

Other calculations include:

finding the interior of polygons in 2 space

rotation and translation operations

transformation of polygons in 3 space into the z=0 plane or 2 space

Distance Calculations:

Point to point calculations are done as follows:

in 2 space

A = Xa,Ya

B = Xb,Yb

$$dist = ((Xa-Xb)^2 + (Ya-Yb)^2)^{1/2}$$

A__________B

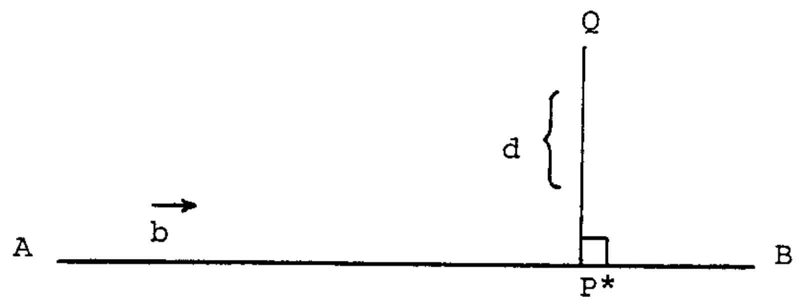
in 3 space

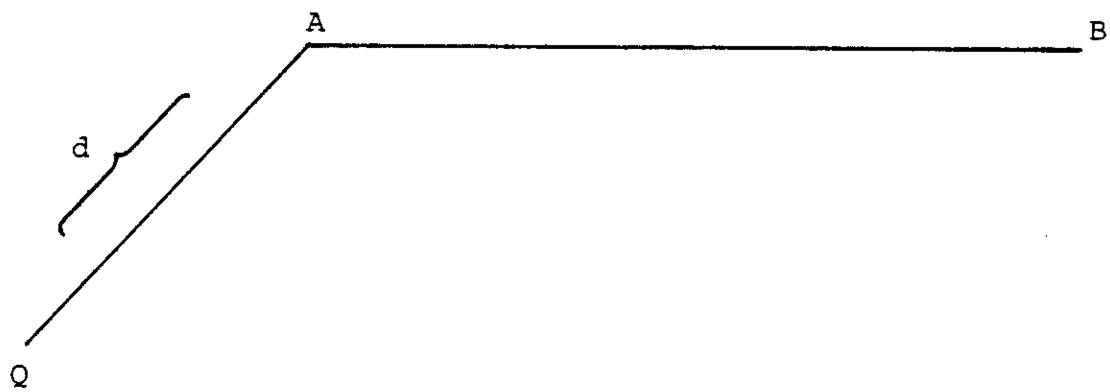
A = Xa,Ya,Za

B = Xb,Yb,Zb

$$dist = ((Xa-Xb)^2 + (Ya-Yb)^2 + (Za-Zb)^2)^{1/2}$$

Point to line or specifically point to line segment distance calculations may be handled the same in 2 space as in 3 space. In general, point to line calculations may be done using relatively simple formula in both 2 and 3 space. In the more specialized case of point to line segment distance calculations the formulas become more complex. For example, given a point Q find its distance from a line segment AB. In the general case d is equal to the perpendicular distance from the segment to the point (see Fig.). In more specialized cases d is equal to the distance from the segment's endpoint to Q (see Fig.). Given these two cases of the problem a general formula for its solution is as follows.





$A = Xa, Ya, Za$

$B = Xb, Yb, Zb$

$Q = Xq, Yq, Zq$

$b = A - B$

$Q - P^* \perp b$

$(Q - P^*) \perp b = 0$

in general the parameter t* locates P* on the line AB; in the case of a segment,

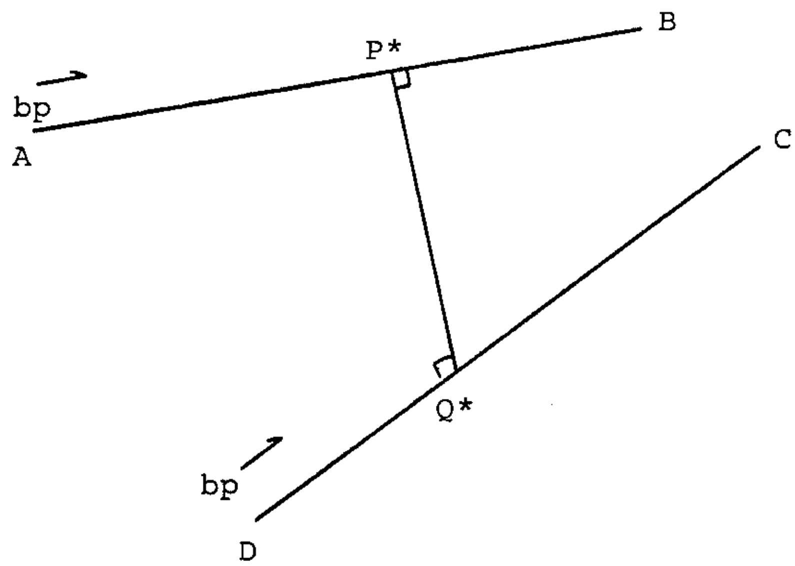
P* = A + t*b if Ø <= t* <= 1

P* = A if t* < Ø

P* = B if t* > 1

t* may be calculated using t* = b . (Q - A)/(b.b) The distance is then the distance from Q to P*.

Line to line distance calculations are similar to point to line calculations in that there is the general case of line to line distance and the more specialized case of line segment to line segment distance calculations. Given two line segments AB and CD the distance d is defined as the perpendicular distance from AB to CD. In other special cases the distance d is the distance from an endpoint to the line AB or CD.



bp = A - B

bq = C - D

P* = A + bp t*

Q* = D + bq s*

bp . (P* - Q*) = 0

bq . (P* - Q*) = 0

given this t* = (-(α*N) + (β*M)) / ((N*L) - M2)

where

L = bp . bp

M = bp . bq

N = bq . bq

α = (bp . D) - (bp . A)

β = (bq . D) - (bq . A)

if 0 <= t* <= 1 then P* = A + bp t*

if t* < 0 then P* = A

if t* > 1 then P* = B

given the P* is found the distance is then calculated as a point to line segment problem.

Point to plane distance calculations are done simply by taking the dot product of a point and the vector representing the plane (see Fig.). The plane vector is calculating the normal vector, normalizing it and calculating the fourth term as the dot product of a sample point and the first three terms in the normal.

$$A = (Xa, Ya, Za, 1)$$
$$P = (Ap, Bp, Cp, Dp)$$
$$dist = |A \cdot P|$$

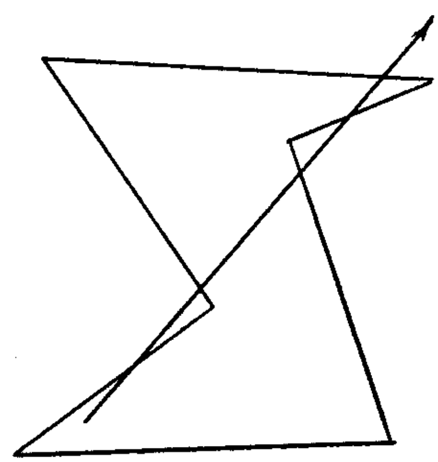
given a plane defined by B,C,D
the plane vector is calculated as follows:

$$B = (Xb, Yb, Zb)$$
$$C = (Xc, Yc, Zc)$$
$$D = (Xd, Yd, Zd)$$

the first three terms of the vector
are $Ap, Bp, Cp = (B - C) \times (C - D) / \|(B - C) \times (C - D)\|$
$Dp = - ((Ap, Bp, Cp) \cdot D)$

Other calculations:

Finding the interior of a polygon in 2 space is a problem of defining a point and then generating a line from that point to some "infinite" point. On defining this ray, a calculation is done to find how many times the ray intersects the polygon. If the polygon is intersected an odd number of times the endpoint is within the polygon, if there are an even number of intersections then the endpoint is exterior to the polygon. (See Fig.)



Transformations are done using 4 X 4 matrices. The matrices for the four basic operations are as follows:

Translation:

$$[x' \; y' \; z' \; 1] = [x \; y \; z \; 1] \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{matrix}$$

Rotation about the x axis

$$[x' \; y' \; z' \; 1] = [x \; y \; z \; 1] \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Rotation about the y axis

$$[x' \; y' \; z' \; 1] = [x \; y \; z \; 1] \begin{matrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Rotation about the z axis

$$[x' \; y' \; z' \; 1] = [x \; y \; z \; 1] \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation of a plane into 2 space is done by first translating the first vertex to the origin. Then rotating the first side about X into the X-Y plane. Then rotating the first side about Z to the X axis. Once this has been done a rotation about X of the second side will bring the remainder of the polygon into the Z = Ø plane.